HEAVY METALS FOR ROBOT BUILDING • AUTOFLEX SUCCESS AT FIRST

SERVO MAGAZINE

FOR THE ROBOT EXPERIMENTER
www.servomagazine.com
MAY 2006

NEW Combat Zone Section pg 44

SUPER ROBONOVA!

Robonova Gets His Own Exosuit

- How to Drive Stepper Motors
- da Vinci Surgical Robot

ROBOLYMPICS

www.robolympics.net

San Francisco, California: Fri-Sun, June 16-18, 2006

Robots from around the world invade San Francisco for the Third Annual International ROBOlympics!

Join us in 2006 for the only international combined-event robot competition. Compete in any of 60 different competitions, including no-cost junior league events (for kids 17 and under.) At the 2005 event, over 650 engineers from 13 different countries competed with more than 450 robots! No matter which event you compete in, you'll still meet hundreds of robot builders from around the world, see new robots you never knew existed, and learn more about robots than you imagined possible!

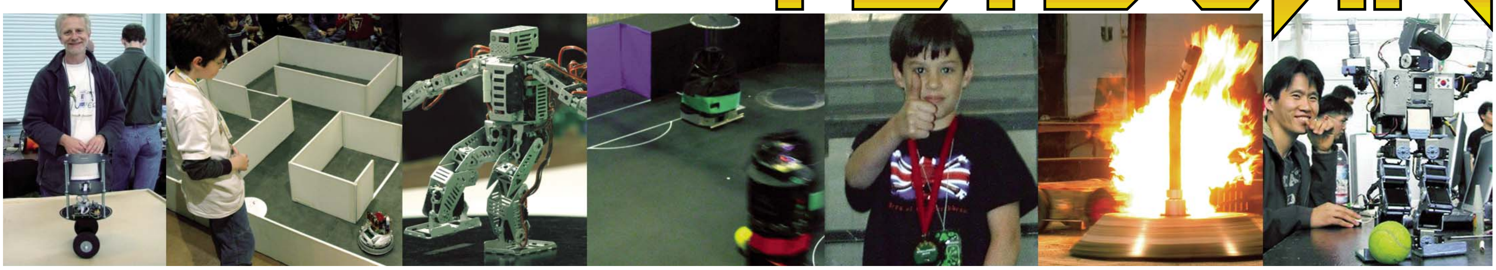
Don't miss this show!

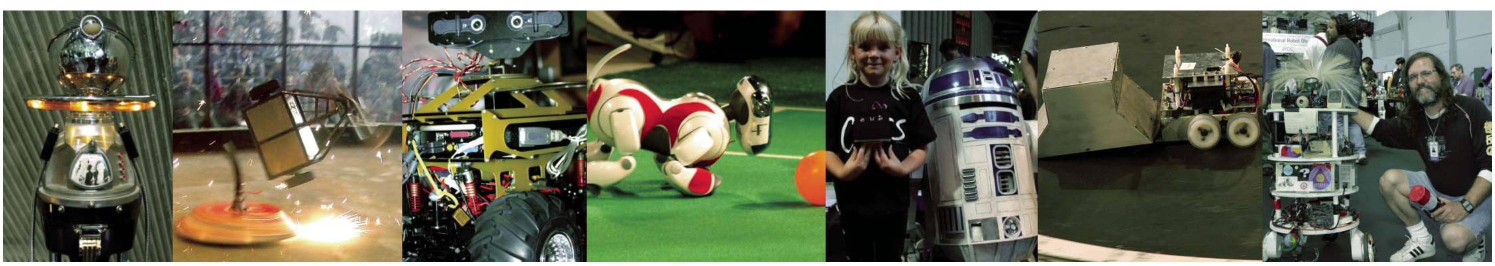
Sponsorship opportunities are still available for the 3nd Int'l ROBOlympics! Take advantage of the tremendous publicity potential! Robotics is becoming the new lifestyle sport of the thinking age. The Renaissance had music, we have DC motors. ROBOlympics is the channel to a growing audience with both a voice and a high disposable income. Contact us for info on this exciting opportunity: sponsors@robogames.net

Events at ROBOlympics:

Combat: 150g-340lbs and 2 autonomus classes;
Robot Soccer: Humanoid, MiroSot, NanoSot, RoboSot, Aibo, Small League, Middle League;
Sumo: 25g - 3kg;
Open: Robot Triathlon, Ribbon Climber, Line Slalom, Mindstorms Challenge, Best of Show, Maze Solving, Aibo Performer, Balancer Race, Fire Fighting, Mindstorms Open, Table Top Navigation, Vex Open, Vex Challenge, Biped Race, Walker Challenge, Robomagellan
BEAM: Speeder, Photovore, Robosapien Hacker
Junior League: Lego - Woots and Snarks, 120 lb Combat (Jr), Mindstorms Challenge, Best of Show, 500 g Sumo, Handy Board Ball, Mindstorms Open, BasketBall Challenge, Vex Open, Vex Challenge
Robo-One: Wrestling, Agility, Stair Climbing, Door Opening, The Eagle, Toss
Art Bots: Musical, Static, Kinetic, Painting, Bartending
Bot Hockey: 3kg & 12 lb
and of course:

TETSUJIN





The Third International ROBOlympics is sponsored by

Robotics Society of America

COMBOTS

SERVO MAGAZINE

SERVO

Features & Projects

23 **AutoFlex Success**
by Michael Wittman
FIRST Team 1714 utilizes Brian Cieslak's programming from his March 06 article and takes home the Rookie All-Star Award.

26 **MICROMOUSE 2006**
by R. Steven Rainwater
The APEC 2006 event offers a unique mix of speeding mice and lint rollers.

30 **Mobility to the Maxx**
by Chris Cooper
Part 2: Controlling the E-Maxx with a PC and joystick.

35 **Demonstrations of Multipath Interferences**
by Ralph D. Lorenz
The Huygens Probe on Titan and a simple laboratory experiment.

39 **BOOK REVIEW**
by Tom Carroll
Robots: From Science Fiction to Technological Revolution.

51 **Build an Advanced Two DC Motor Test Platform**
by Daniel Ramirez
Use this low-cost platform to develop and evaluate various PID control algorithms using DC motors for robotics applications.

63 **ComBots Cup and RFL National Championship**
by Dave Calkins
Sewer Snake takes home the $10,000 prize.

ON THE COVER

SUPER ROBONOVA gets his own exo-suit!

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. APPLICATION TO MAIL AT PERIODICALS POSTAGE RATE IS PENDING AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

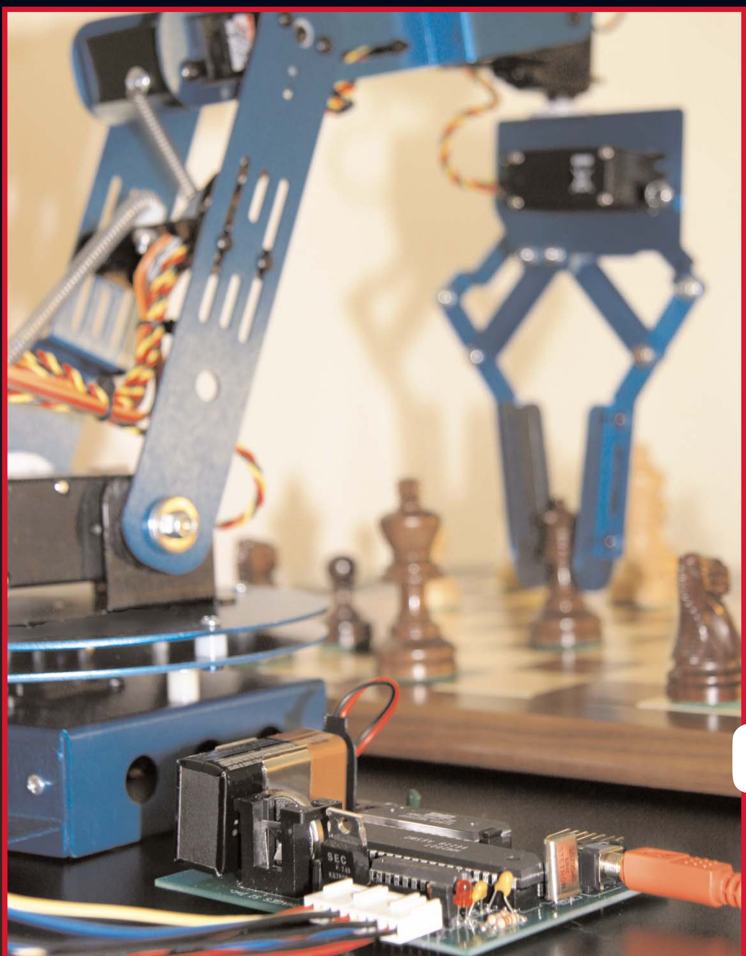
05.2006
VOL. 4 NO. 5

Departments

06 Mind/Iron
07 Bio-Feedback
28 Events Calendar
29 Robotics Showcase
42 Menagerie
50 New Products
84 *SERVO* Bookstore
88 Robo-Links
90 Advertiser's Index

NEW MONTHLY SECTION!

44 The Combat Zone



Columns

08 Robytes *by Jeff Eckert*
Stimulating Robot Tidbits

10 Ask Mr. Roboto *by Pete Miles*
Your Problems Solved Here

14 GeerHead *by David Geer*
da Vinci Surgical Robot

19 Rubberbands and Bailing Wire *by Jack Buffington*
How to Drive a Stepper Motor

67 Twin Tweaks
by Evan and Bryce Woolley
Super Robonova

72 Programmable Logic
by Gerard Fonte
Small- to Medium-sized Programmable Logic Parts.

76 Robotics Resources
by Gordon McComb
Heavy Metal for Robot Building

81 Robot Trends *by Dan Kara*
Truisms (or not!) of the US Robotics Market

86 Appetizer *by Alex Brown*
Group Robot Projects

89 Then and Now *by Tom Carroll*
The UMI R-Theta Robot

Coming 07.2006

Voice Recognition for Robotic Control

Mind / Iron

by Kevin Berry

SERVO Magazine — a long time supporter of combat robotics — initiates monthly coverage of the sport in this issue. By bringing in articles from the best builders and competitors, it is hoped to encourage readers to attend an event as a spectator or (better yet) a builder, getting many more involved in this growing and exciting sport.

Whenever I talk to someone about my hobby, I inevitably get the same question — "Didn't that die when the TV shows went off the air?" My answer always surprises them. The sport has mushroomed since then; with grass roots events springing up all over North America (pardon the mixed metaphors!) There are now about 50 events a year, and more builders than ever.

What has changed is the number of weight classes and distribution of bots. In the glory days of television coverage, there were just four weight classes, from 60 pound "lightweights" to 340 pound "superheavyweights." Today, Robot Fighting League events commonly host 10 weight classes, with the new ones from 150 grams to 30 pounds. These smaller weight classes, due to their lower cost and easier transportability, are proving extremely popular.

UI Productions Builders Database (**www.buildersdb.com**) lists almost 2,700 builders and 3,700 combat robots. Admittedly, some of these are retired, and others are "vapor bots" with the hardware still a dream hovering over someone's empty piggybank. Still, the numbers clearly show the shift towards smaller bots, with 72% being 30 lbs or under. Experienced Event Organizers are finding it difficult to attract the "big boys" to more than one or two events a year, while local, small bot tournaments with 15 to 30 entries flourish.

What good can come from a sport where hundreds (or thousands) of dollars of hardware and hundreds of hours of work can be ruined in under three minutes? A recent experience illustrates the value of the sport to me. My 12-year-old daughter — a combat veteran with a dozen events under her belt — is on a student Odyssey Of The Mind team. They had to build a human powered vehicle, develop a skit, and produce two remotely triggered, self operating "technical elements." The team based their skit on The Three Stooges, and had the idea of animating mannequins to slap heads and poke eyes as their technical elements.

The other middle schoolers were stumped. My daughter found a marvelous website — **www.flying-pig.co.uk/mechanisms/** — which shows how to convert between various forms of motion. Pretty soon she asked "Dad, can I tear apart my robot?"

She was referring to a four servo pusher antweight that had long been outclassed and retired. She quickly assembled a hacked servo, battery pack, power switch, scrap metal, cardboard, and screws, and had a functioning animated arm. After some tweaking, it was installed on their props and off they went to the competition. The judges asked about the technical element. Dressed in her costume — a pink prom dress and

Mind/Iron Continued →

Published Monthly By
T & L Publications, Inc.
430 Princeland Court
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Product Order Line **1-800-783-4624**
www.servomagazine.com

Subscriptions
Inside US **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

**ASSOCIATE PUBLISHER/
VP OF SALES/MARKETING**
Robin Lemieux
display@servomagazine.com

CONTRIBUTING EDITORS
Dave Prochnow, Gerard Fonte, Pete Miles, David Geer, Jack Buffington, R. Steven Rainwater, Chris Cooper, Michael Wittman, Ralph Lorenz, Tom Carroll, Kevin Berry, Steve Judd, Daniel Ramirez, David Calkins, Evan Woolley, Bryce Woolley, Gordon McComb, Dan Kara, Alex Brown, Peter Smith, Jeff Eckert, John Kruse

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@servomagazine.com

WEB CONTENT/STORE
Michael Kaudze
sales@servomagazine.com

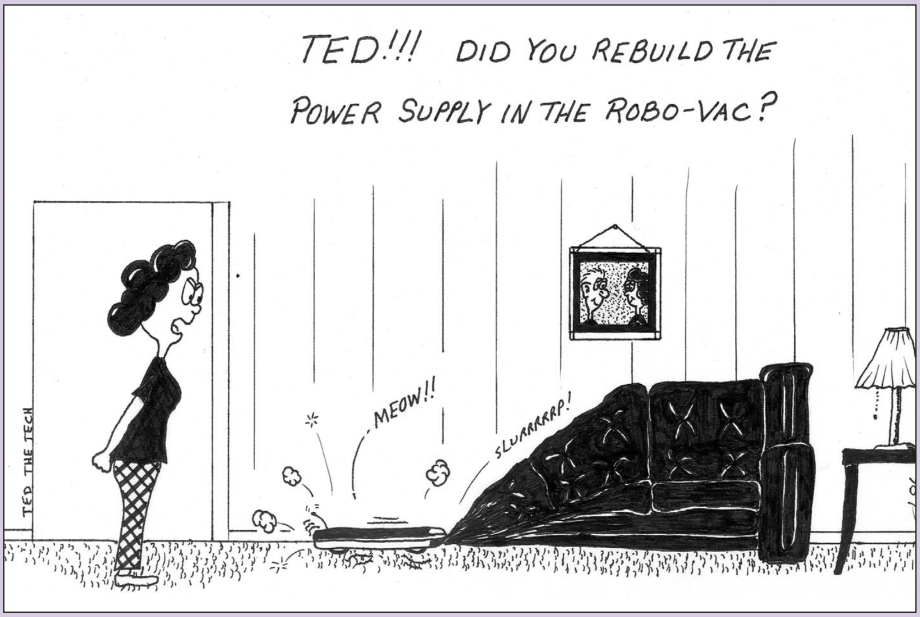
PRODUCTION/GRAPHICS
Shannon Lemieux
Michele Durant

Biped Bonanza

SERVO Magazine readers will soon be able to rejoice — stores will soon be filling with biped options. Since the release of WowWee Robotics Robosapien, robot manufacturers have been literally throwing out wheels and tracks in favor of feet.

One of these new bipeds on the block is the Internet Renaissance (ITR) robot from Speecys. Using Robot Transaction Markup Language (RTML) for communication via the Web, ITR sports an exciting boatload of features including, 168 LEDs, USB, Wi-Fi, miniSD Slot, RPU-50 CPU, and an OS built on NetBSD. You can learn more about ITR at: **www.speecys.com/itr**

BIO-FEEDBACK



tiara — she proceeded to give them a *SERVO* quality lecture on hacking servos, the advantages of NiMH batteries, fulcrum positioning on Class 1 vs. Class 3 levers, and the speed vs. torque tradeoff problem. Eyes glazed, they awarded her team first place.

This is why I love combat robotics.

The sport brings together a unique crowd of technical experts, students, families, and R/C buffs in an atmosphere that's supportive in the pits and brutal in the arena, creating a learning atmosphere that has a high "cool" factor and tons of practical technical opportunities. **SV**

Robytes

by Jeff Eckert

Are you an avid Internet surfer who came across something cool that we all need to see? Are you on an interesting R&D group and want to share what you're developing? Then send me an email! To submit related press releases and news items, please visit www.jkeckert.com

— Jeff Eckert

UUVs Launched and Recovered by Sub

The USS Scranton employs UUV systems for mine sweeping. Photo courtesy of the US Navy.

Earlier this year, the fast-attack submarine USS Scranton successfully demonstrated the homing and docking of an unmanned undersea vehicle (UUV) system during at-sea testing. The two UUVs used in the testing are a part of the AN/BLQ-11 Long Term Mine Reconnaissance System (LMRS), which was designed to enable submarines to conduct clandestine undersea surveys to locate mines.

The UUVs are launched from the sub's torpedo tubes, after which they are controlled via an underwater acoustic communication system. After the UUV is launched from the submarine's torpedo tube, it transits to a series of preprogrammed waypoints. Meanwhile, the submarine maneuvers to rendezvous with the UUV. Homing and docking sonar guides the UUV toward a recovery arm, which is a unique docking mechanism that extends out of the ship's upper torpedo tube. After the UUV is captured, the recovery arm guides the UUV into the lower torpedo tube and back into the submarine.

Ultimately, the goal is to create a version that can act autonomously and act as a tool for clandestine intelligence, surveillance, and reconnaissance (ISR) operations, thus providing a wealth of information about battlespace areas.

World's Fastest Picker?

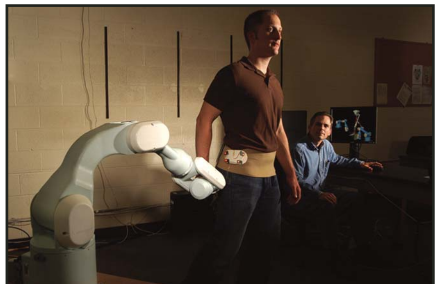
The Quickplacer bot is claimed to be the world's fastest. Photo courtesy of Fatronik.

Launched recently at the International Machine-Tool Biennial, in Bilbao, Spain, was the Quickplacer robot built by Fatronik. It's claimed to be the world's fastest, although we're not talking about ground speed here. It's really a stationary industrial handling machine consisting of four coordinated actuators. It has four degrees of freedom with displacements along three translations, and it rotates on its vertical axis.

It is basically a cylinder with a diameter of 1,200 mm and a height of 250 mm. The bot's rotational capacity covers ±200°, which enables it to position an object in any orientation, and it is guided by a vision system (either black-and-white or color). It is capable of pulling up to 15 G of acceleration, can pick up more than 200 items per minute, and can even grab them from, or place them on, a moving conveyor belt.

Quickplacer is designed to handle items of various sizes and shapes, weighing up to 2 kg. Suggested applications include positioning chocolates in boxes, quality control in vegetable processing, packaging of lipstick, and feeding various products into packaging machines. You can find out more about the company and its products at **www.fatronik.com** but only if you read Spanish.

X-Ray Robot Under Development



This UF-designed robot performs on-the-fly x-ray diagnoses. Photo by Kristen Bartlett, courtesy of the University of Florida.

Somewhere in the neighborhood of eight million people per year are hospitalized for musculoskeletal conditions or injuries, and most conditions are diagnosed using x-rays, MRI, or CT scans. Although these techniques can be effective, they don't work well with injuries that show up only when a joint is in motion, such as damage to a kneecap and shoulder. Surgeons sometimes have to operate to diagnose these and other injuries, which can lead to unnecessary surgeries.

However, a University of Florida (**www.ufl.edu**) engineer is working on a robot that is intended to shadow and shoot x-ray videos of injured people as they walk, climb stairs, stand up from a seated position, or pursue other activities. The photo shows how the robot is intended to follow a patient's movement by tracking an LED-lit patch attached to his thigh.

In this demonstration, the robotic hand was just carrying a standard video camera, and it didn't actually

have the tracking accuracy needed to generate a useful video x-ray. But Scott Banks, chief researcher on the project, has applied for a $275,000 grant from the National Institutes of Health to allow him to perfect the concept. UF has also applied for a patent on the new imaging technique, and Banks says it's possible that it could become standard equipment in hospitals.

Bot Kit Priced Below $100

If your bank account is looking a little depleted but you still want to get started on building a mechanical critter, the Chicago Area Robotics Group (**www.chibots.org**) has a deal for you. Its ChiBots Alpha (CBA) robot has been in development and testing by club members for two years and is now available from Mike Davey through BudgetBots.com. Mike describes himself as "just a guy in his garage, with lots of help from others, and with a mission."

Conceived as a low-cost standard platform to help club members get started with robotics, the kit is designed to be suitable for beginners, but flexible and expandable enough for a more experienced builder and programmer. The robot is controlled by its on-board BASIC Stamp® 2e or 2sx, which features protected I/O lines. A 60-page manual steps you through soldering and testing the main board, converting the RC servos to continuous rotation, and assembling the chassis, and all you need are basic electrical and hand tools.

The manual includes a section on programming, and you also get sample programming software on CDs. Wheel encoders and a flexible line-following module are also available as kits, and other add-on modules are being developed. The basic kit will run you $95 with the 2e, and the 2sx version is $10 more. Volume discounts are also available. For complete information, go to **www.budgetbot.com**

DARPA Looking for Bot Bugs

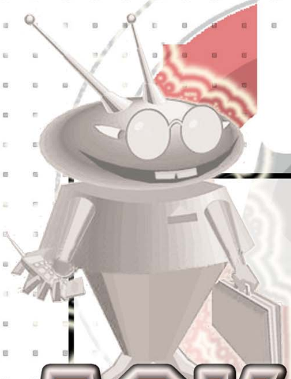
If you are up for a serious project, please note that the Defense Advanced Research Projects Agency (DARPA) recently posted a notice (see **www.darpa.mil/baa/baa06-22.html**) in which it solicits research proposals in the area of hybrid insect MEMS. The agency specifically excludes anything that is based primarily on the existing state-of-the-art, so make it original.

Basically, what they want is insect-cyborgs that are created by integrating microsystems into the bugs during their early stages of development to yield a "more reliable bio-electro-mechanical interface to insects, as compared to adhesively bonded systems to adult insects. Once these platforms are integrated, various microsystem payloads can be mounted on the platforms with the goal of controlling insect locomotion, sense local environment, and scavenge power."

While DARPA prefers flying insects, you might get funding for something that only hops or flys. But it "must also be able to transmit data from DOD relevant sensors, yielding information about the local environment. These sensors can include gas sensors, microphones, video, etc." The ultimate goal is to develop a swarm of cyberbugs that can land within 5 meters of a specific target from a distance of 100 meters. The mind boggles at the possible applications.

Conference on Robotics and Automation

If you read this issue soon after it arrives, it won't be too late to attend the 2006 IEEE International Conference on Robotics and Automation, which will be held May 15 through 19 at the Hilton in the Walt Disney World Resort, in Orlando, FL. The theme of the ICRA 2006 conference is "Humanitarian Robotics," by which they refer to the use of robotic technology in areas such as search-and-rescue missions, homeland security, humanoid robots, personal and service robots, mine removal, and so on. For details, visit **www.icra2006.org** SV



Our resident expert on all things robotic is merely an Email away.

roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by **Pete Miles**

Q. Is there an easy way to remove a broken tap? If I can get a pair of pliers on the broken tap, I can usually work it loose. If I can't, I end up having to remake the part. Sometimes this is no big deal, but other times I don't want to redo all the work I already put into the part. So I am curious to know if there is an easy way to remove a broken tap.

— Mike Montgomery

A. Well, the short answer is no. For the most part, there really isn't any easy way to remove a broken tap. The best thing to do is not to break the tap in the first place. But that is easier said than done. Here are the three tools that I have used to remove broken taps: 1) a tap extractor, 2) abrasive waterjets, and 3) electro-discharge machining. Most people don't have access to abrasive waterjet or electro-discharge machines, and it can be expensive to contract out this work. That leaves a tap extractor.

I have used the three- and four-flute tap extractors from Walton (**www.waltontools.com**) with some success (see Figure 1). They are easy to use. Just push the fingers of the extractor down the hole along the sides of the broken tap, then slide the sleeve down flush with the workpiece/tap. With the tap handle, wiggle the extractor until the broken tap is loosened and then unscrew the tap.

The reason I say "with some success" is because, when the tap is very tight, the fingers of the tap extractor may break when trying to remove it. If that happens, you will either have to remake the part or get an abrasive waterjet or electro-discharge machine to cut the tap out of the hole.

Figure 1. Walton tap extractor.

Dealing with broken taps can be a frustrating experience. When I break a tap, I often do a really good job at it and end up breaking a lot of tap extractors (along with a lot of other tools) trying to remove the broken tap. That's why — although the Walton tap extractor is a fine tool, and I highly recommend it — the short answer to whether there's an easy way to remove a broken tap is "no" and the best thing to do is to not break taps in the first place.

The top 10 reasons taps break are (not in any particular order):

1) Using one hand to turn the tap handle.

2) Turning the tap more than 1/4 turn before back-turning to break the chips.

3) Forcing the tap to turn when it is stuck.

4) Not going down the center of the hole (angular misalignment).

5) Using the wrong tapping lubricant for the material being tapped.

6) Using an undersized tap drill (it is better to drill a few thousands of an inch too big than to drill too small).

7) Not pulling the tap out of the hole several times to remove all the broken chips.

8) Bottoming out the tap in a blind hole.

9) Bumping the tap handle with the hand or arm with the tap in the hole.

10) Using a dull tap.

Anyone who taps a lot of holes will likely break a tap for each of these

reasons. The real culprit is usually being in a rush and not paying attention or being cheap and reusing dull taps that should have been tossed out years ago. Though a tap may still look sharp, if it begins to feel "sticky," even with plenty of lubricants, the tap is dull and should be replaced.

Hopefully this will help you deal with the exciting world of tapping holes.

Q. Do you know of any inexpensive methods, other than encoders, to monitor the speed of a motor?

— Joe Konoske

A. Try taking a look at Hall-effect sensors. They are both inexpensive and easy to integrate into speed-monitoring systems. Hall-effect sensors are small semiconductors that are very sensitive to changes in magnetic fields. They can be used to measure the intensity of a magnetic field or to monitor changes in a magnetic field. These sensors make ideal magnetic switches for counting gear/sprocket teeth or monitoring the number of times a magnet passes in front of the sensor.

The Melexis MLX90217 Hall-effect (**www.melexis.com**) sensor sold by Parallax (**www.parallax.com**) for $4.25 each, has internal electronics that convert changes in magnetic fields into a digital signal that can be used to count the number of times a magnet passes in front of the sensor. All of this is mounted inside a small TO-92, transistor-sized package. Figure 2 shows a photograph of this sensor and Figure 3 shows the manufacturer's minimum recommended protection circuit for using this sensor. As this figure shows, only two additional components are needed to use this sensor: a resistor and a capacitor.

When using this sensor, the North-South pole orientation of the magnet is critical for proper operation. The magnet's orientation determines whether it should be located in front of or behind the sensor.

The simplest implementation of this sensor is attaching a magnet to a moving surface that will periodically pass in front of the sensor. Each time the magnet passes in front of the sensor, the output will change from a High to a Low and back to a High state. While the magnet is directly in front of the sensor, the output will be in a Low state. In this configuration, the South pole of the magnet must be facing the front face of the sensor. Figure 4 illustrates how the output state of the sensor changes as the magnet passes in front of the sensor. In this figure, the front face of the sensor is the face with the printed text and side bevels.

Another implementation for speed monitoring is counting the number of teeth of a steel gear or sprocket that passes in front of the sensor. *Note*: the gear or sprocket *must* be a magnetic material (i.e., steel or some other material that attracts magnets) for this method to work. In this application, a magnet is glued to either the back or front side of the sensor. Then, when a gear/sprocket tooth passes in front of the sensor, the output will change from a High to a Low and back to a High state. Figure 5 shows an illustration of how this works.

In this configuration, if the magnet is to be glued to the front face of the sensor, the South pole of the magnet must be in contact with the front face. If the magnet is to be glued to the back of the sensor, the North pole of the magnet must be in contact with the rear surface of the sensor.

The next question that comes up is how to determine the magnetic-pole orientation of the magnet, since most magnets are

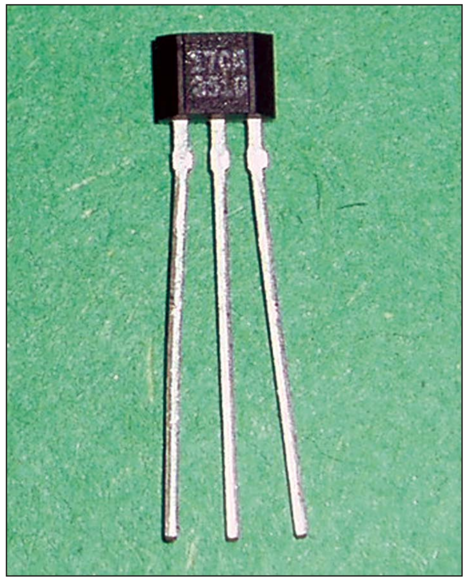


Figure 2. Melexis MLX90217 Hall-effect sensor.

Figure 3. Minimum recommended protection circuit for the MLX90217 Hall-effect sensor.

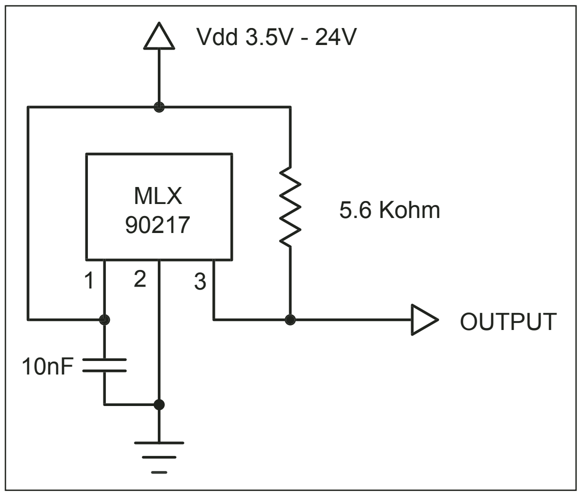
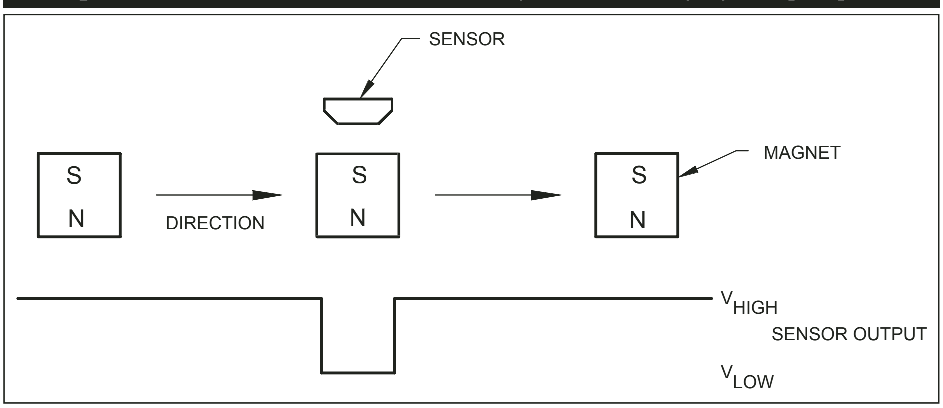


Figure 4. Illustration of how the sensor's output is affected by a passing magnet.



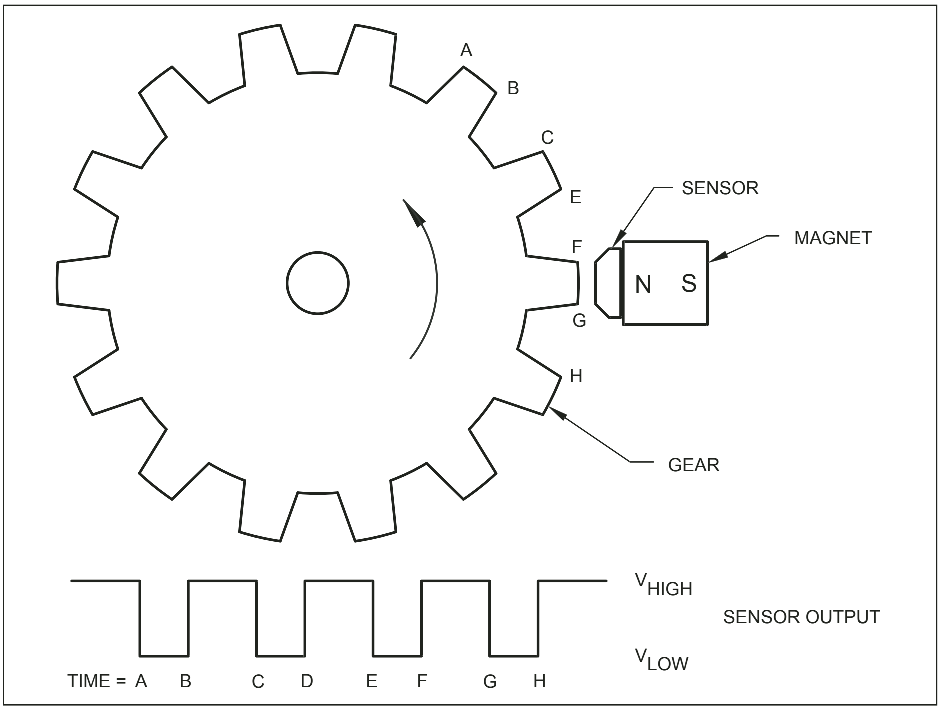


Figure 5. Illustration of how the sensor's output is affected by a passing gear tooth.

not marked. This sensor can also be used to determine the magnet's polarity. To do this, hook up the output to either a multimeter or an oscilloscope and watch how the voltage changes as the magnet passes in front of the sensor. Take the magnet and pass it in front of the sensor going slowly from left to right, then reverse the direction and go right to left. If the output changes from High to Low to High in both directions, then the side facing the sensor is the South pole. If the output stays Low after either pass across the front of the sensor, then a North pole face was the last face the sensor saw. Change the orientation of the magnet a few times and repeat the process. When you find the orientation for which the output stays Low as you pass the magnet in either direction, you'll know that the side of the magnet facing the sensor is its North Pole.

Another thing to keep in mind is that the sensing distance is a function of the intensity of the magnetic field. Larger magnets will have a greater sensing range than smaller magnets. Rare earth magnets such as neodymium have greater sensing ranges than common, inexpensive ceramic magnets. My 1/8 in^2 neodymium magnets have a sensing range of about 0.3 inches. My 1.5 inch, "C" shaped neodymium magnets have a sensing range of about 0.8 inches when the magnet is passing in front of the sensor. In the case of the sensor with a magnet glued to its face to sense a moving gear tooth, the sensing distances are about half this. These distances will vary depending on the magnets you use.

To determine the speed only

Listing 1

```
Hall_Effect Demo
'This demo program illustrates how to use a Hall
'Effect sensor to monitor gear/motor speed.
'The sensor output is connected to Pin 0 on the
'Basic Stamp.

Period    CON  3484   'The following period values
                      ' represent the time base for sampling for one second.
                      '   Use 3484 FOR BS2P & BS2PX
                      '   Use 1389 for BS2pe
                      '   Use 2500 for BS2sx
                      '   Use 1000 for BS2 & BS2e
N         CON     1   'Number of teeth on the gear or number of magnets on the diameter of the wheel/gear.
T_Sample  CON     1   'Number of seconds to sample.

tmp       VAR  Word   'Tempory variable for storing the number of sensor counts per Period.
RPM       VAR  Word   'Rotational speed of the gear/wheel in revolutions/minute

Main:
  COUNT 0, T_Sample*Period, tmp
  RPM = tmp * 60 / T_Sample / N
  IF RPM < 1 THEN
    DEBUG "Stopped", CR
  ELSE
    DEBUG "RPM = ", DEC RPM, CR
  ENDIF
  GOTO Main
```

requires counting the number of times the magnet or number of gear/sprocket teeth passes the front of the sensor over a given time period, such as, say one second. Then divide that result by the number of teeth the gear/sprocket has and multiply by 60 to get RPM. If you are working with relatively slow-moving parts or large-diameter parts, you may want to use more magnets to get better resolution or sample for a longer time period. The equation below shows how to calculate the speed of a sprocket/gear/wheel/etc.

$$\text{RPM} = \frac{\textit{Number of Counter per Second}}{\textit{Number of Teeth/Magnets}} * 60$$

With a BASIC Stamp 2, only two lines of code are needed to make the measurement and calculate the result, which shows how easy it is to implement this type of a sensor using a microcontroller.

```
Count Pin0, 1000, tmp
 RPM = tmp * 60 / Tooth_Count
```

The program shown in Listing 1 is a simple BASIC Stamp program utilizing this sensor to measure the rotational speed of several gear motors. As you can see, there is not a lot programming required to use this sensor.

Figure 6 shows a photograph of two different motors with tiny neodymium magnets attached to the end of the shaft. The Graupner 500E motor is a 12 V motor with a no-load speed of 12,000 RPM. With this sensor, I was able to track the speed of this motor to at least 16,000 RPM. I didn't test at speeds above this because magnetic attraction was the only thing holding the sensor onto the shaft, and I was worried it might fly off.

Because the Lynxmotion planetary gear motor was a low RPM motor, I added five magnets to the shaft to get a more accurate resolution. Here, I changed the variable "N" in the program to have a value of five (for the five magnets). At 12 V, the output speed was 72 RPM. If I had used only one magnet on this motor, the sensor would have indicated that the motor was turning at either 60 RPM or 120 RPM, depending on whether that magnet had passed by once or twice during the one-second interval. SV

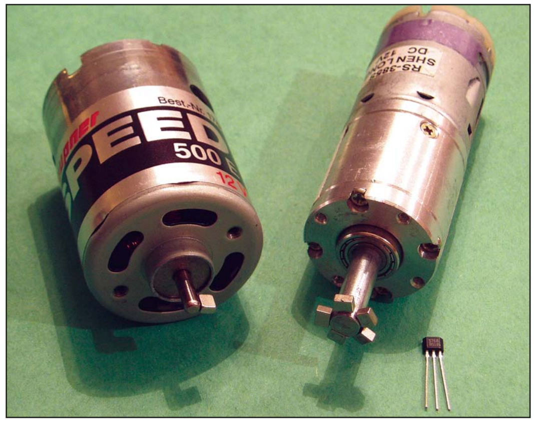


Figure 6. Two different 12 V motors showing magnet placement for speed measuring, along with the Hall-effect sensor for size comparison.

GEER HEAD

by David Geer
Contact the author at geercom@alltel.net

da Vinci Surgical Robot

The Doctor Will See You Now

You're Cutting What, Down Where, Using a What?!

A new robotic-assisted surgery technology is in use across the US and around the world to aid doctors in performing safe, accurate, and much less invasive procedures in the most sensitive of places.

Yes, not only has the FDA approved the da Vinci Surgical Robot from Intuitive Surgical, Inc., for things like heart surgery but also for prostate and gynecological operations, as well.

In fact, the list of surgeries for which the technology is available is quite long. Cardiac surgeries such as Coronary Revascularization, Mitral Valve Repair, ASD repair, and Epicardial Pacemaker Lead Replacement are all being performed with da Vinci.

More general procedures the da Vinci system has mastered include Gastric Bypass, Nissen Fundoplication (for acid reflux), and Heller Myotomy (for Achalasia).

Thoracic operations include Esophagectomy (for cancer of the esophagus), Thymectomy (for myasthenia gravis), and Lobectomy (for lung cancer).

And for those sensitive areas — ladies first — gynecologic procedures like Myomectomy (for uterine fibroids) and Hysterectomy. A number of urologic procedures are available for the gents including prostatectomy, pyeloplasty (for ureteropelvic junction obstruction, UPJO), Cyctectomy (for bladder cancer), nephrectomy and partial nephrectomy (for renal cancer), ureteral reimplantation (for vesicoureteral reflux), and vasovasostomy (for fertility).

Typical benefits from da Vinci assisted surgery include faster recovery time, shorter hospital stay, less pain, less scarring, and a quicker return to normal routines.

A few hundred da Vinci surgical systems have already been installed around the world with great success and rave reviews from patients.

According to Javier F. Magrina, M.D., chair, department of obstetrics and gynecology, Mayo Clinic, robotic surgery like that provided by da Vinci is an upgraded form of minimally invasive surgery with major benefits to the patient.

"Robotic operations are more precise than conventional surgery. We have also noted a reduced use of pain medications after robotic surgery, indicating less tissue trauma," says Magrina.

The complete da Vinci surgical system in action.

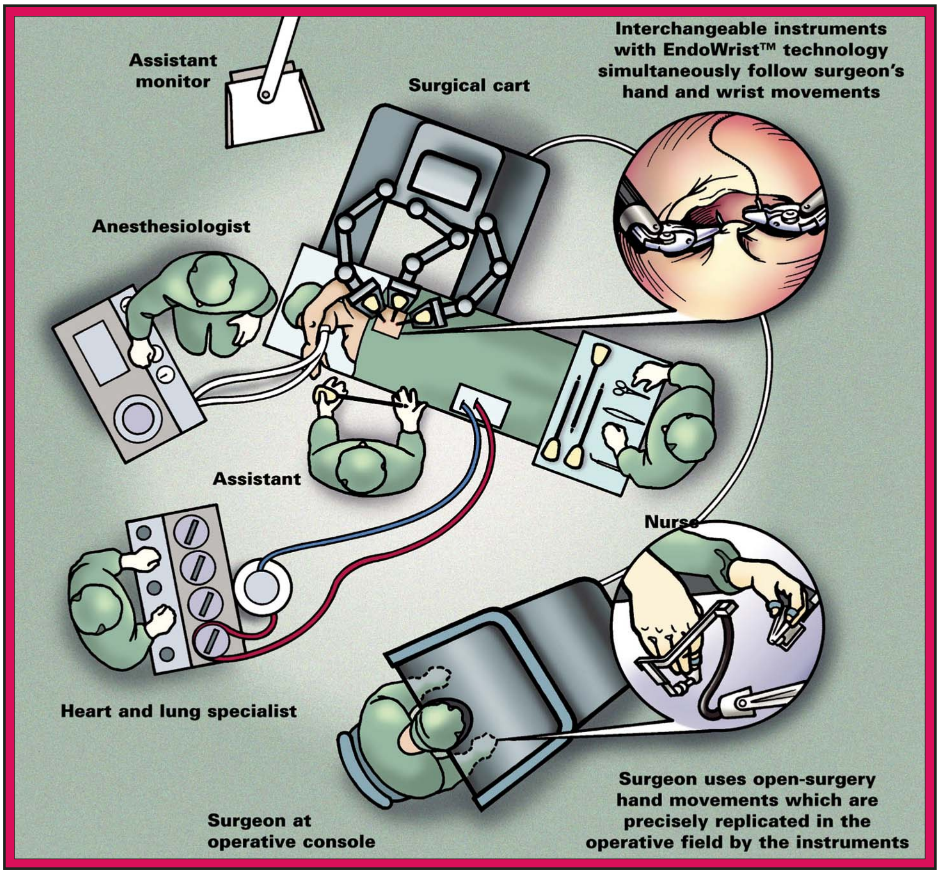
The da Vinci Surgical System

Da Vinci lets surgeons operate

ENDOWRIST GRASPER ROBOTIC SURGICAL MEDICAL INSTRUMENTS

The Cobra Grasper has interlocking teeth enabling it to securely hold dense tissue like fibroid tumors or the prostate. It opens 60 degrees wide. It has serrated inner jaws to handle sutures and needles. The device is intended for grasping and retracting fibroid tumors, pelvic fascial layers, and the prostate.

The Double Fenestrated (window like openings) Grasper employs a double fenestrated jaw, a rounded triangular tooth profile, and a 25 mm jaw length. It enables grasping of bowels, stomach, and the colon. Applications include gastric bypass, and nissen and colorectal procedures.



da Vinci operating room schematic showing all major elements with embedded explanations.

through a series of bodily incisions of only 1-2 cm.

The system includes an ergonomic console for the surgeon to sit at, view the area of the surgery, and manipulate the robotic surgical tools. The console communicates back and forth with a surgical cart at the patient's side, which performs the surgery.

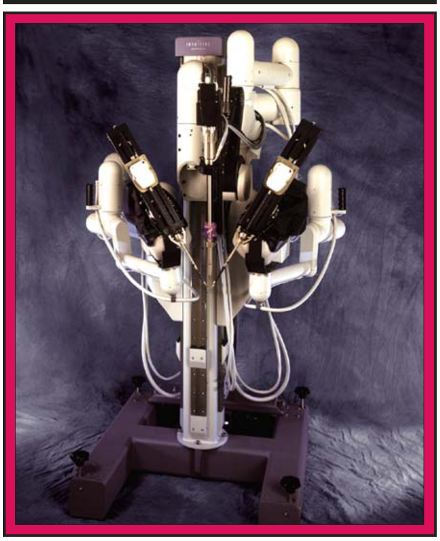
The cart has four robotic surgical arms with several degrees of freedom to posit themselves and their attached EndoWrist operating instruments, as well as a 3D camera.

The da Vinci console and technology translate the surgeon's hand movements into perfectly mirrored movements by the surgical instruments.

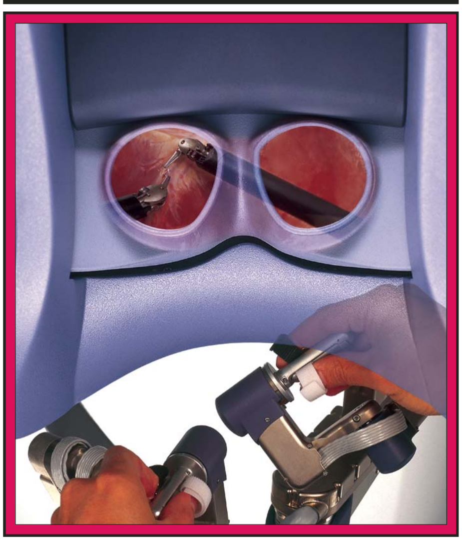
The 3D Intuitive motion vision system is an interface that helps the surgeon feel like he or she is performing a traditional open surgery. Through a stereo 3D viewer, the surgeon sees the patient anatomy in high magnification, vibrant color, and natural depth-of-field of vision.

The surgeon operates using the

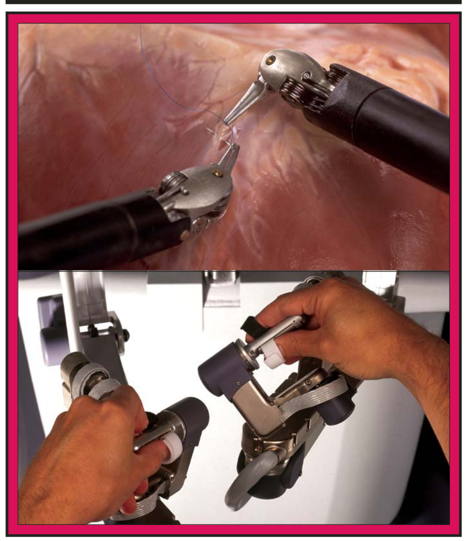
Patient side cart with surgical arms and wrists.

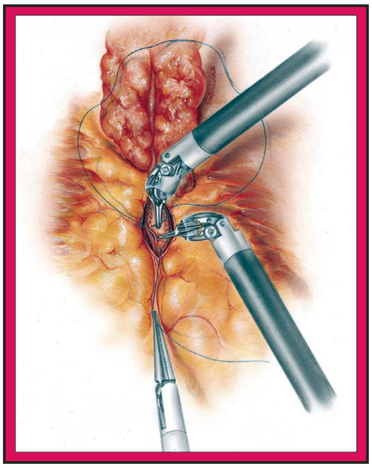


What the surgeon sees and how he operates using the InSite Vision technology.



Surgeon's Console Master controls and endowrists in action.





Close-up of operative field view during a coronary anastamosis.

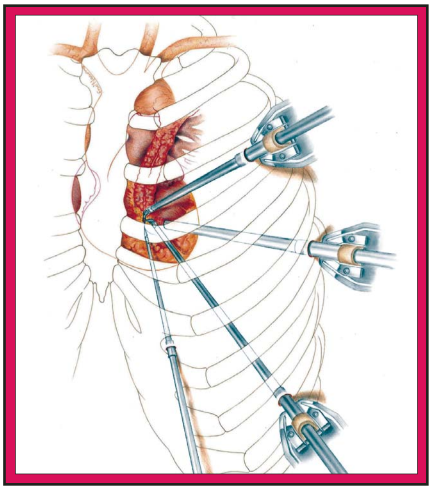


Illustration of minimally invasive ports.

ENDOWRIST SHEARS AND SCISSORS

The Curved Scissors are well-suited to cardiac, urological, and general surgery.

They have a curved jaw for easy access to hard-to-reach tissue by reaching around tissues. They have thin sharp cutting blades and tapered tips for blunt tissue dissection.

The scissors are best for pericardiotomy, atriotomy, and cutting heart valve leaflets, as well as dissection around the prostate, nerve sparring, and transection of the bladder neck, urethra, or DVC.

The Harmonic Curved Shears feature a curved jaw. These shears minimize the thermal spread [heat] and offer precise hemostasis control. They offer less tissue charring than that of electrocautery (a metal cauterizing instrument heated by electricity).

console's master controls. These move the EndoWrist operating instruments, which have jointed wrists that have a wider range of motion than the human hand. By means of motion scaling and tremor reduction, the surgeon's hand movements are refined for more precise surgery.

The system also uses minimal force feedback sensations from the instruments to give the surgeon a useful substitute for tactile sensation.

The system also has redundant fail-safes to insure patient safety.

Clinical studies even point to the possibility of better surgical results from da Vinci, such that cancer, for example, could be better controlled or so that prostatectomy would lead to much lower incidences of impotence or incontinence.

Hardware

The robotic cart can have three to four arms, meaning up to three surgical instrument arms and the endoscope arm. These laparoscopic arms pivot at the 1-2 cm operating ports/incisions. This means the arms don't need to rest on the patient's body wall, which normally would cause tissue damage.

Surgical team members along side of the patient install instruments in the arms as they are needed; they also prep the 1-2 cm incisions and supervise the instruments while in use. Quick-release levers make instrument changes fast during surgery.

The EndoWrist Instruments are several. They have seven degrees of freedom of movement and 90 degrees of articulation to better than replace the capabilities of the human hand and wrist. Each one is uniquely suited to a specific surgical purpose or purposes such as those that are common in surgery like clamping, suturing, and

	Open	Laparoscopic	*dVP*
Patients	100	50	100
Operative Time (Min.)	164	248	140
Blood Loss (mL)	900	380	<100
Cancer Remaining	24%	24%	5%
Complications	15%	10%	5%
Catheter (Days)	15	8	7
Hospitalization (Days)	3.5	1.3	1.2

Menon M. Robotic radical retropubic prostatectomy. BJU Int. 2003 Feb;91(3):175-180

Table 1. Urology: Comparison of Open Prostatectomy, Laparoscopic, and *da Vinci* Prostatectomy.

	2001 STS Nat'l Database Sternotomy MVR	*da Vinci* MVR Multi-center Trial
Patients	893	112
Mortality	2.2%	0%
Major Complications	13.1%	9.8%
Neurological Complications	2.4%	0%
Hospitalization (Days)	8.5	4.7

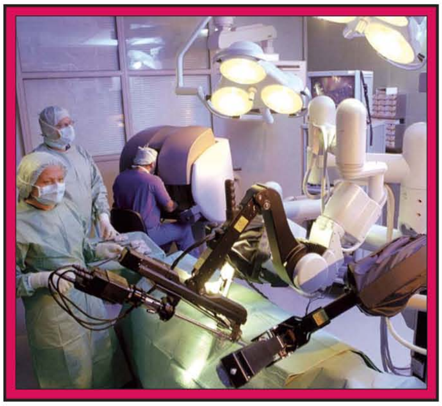
STS and da Vinci multi-center trial data on file. See also Tatooles, A. da Vinci Robotic Mitral Valve Repair: Outpatient Procedure? Presented at the Society for Thoracic Surgeons 39th Annual Meeting, ***www.sts.org/2003webcast/shows/tatooles.html***

Table 2. Cardiac: Comparison of Relevant Clinical Variables for Mitral Valve Repair (MVR).

	Laparoscopic*	da Vinci Heller Myotomy†
Patients	100	54
Mucosal Perforation	8%	0%
Relief from Dysphagia	93%	93%
Post-op GERD (heartburn)	34%	19%
Post-op lower esophageal sphincter pressure (LESP)	22 ± 2 mm Hg	7.1 ± 3.8 Hg
Average Operation Time (min)	110-144 ±7	162 (average of all patients, range = 62-210); 90 post learning-curve — last 10 cases

**Sharp KW, Khaitan L, Scholz S, Holzman MD, Richards WO. 100 consecutive minimally invasive Heller myotomies: lessons learned. Ann Surg. 2002 May;235(5):631-8; discussion 638-9. †Unpublished paper on file, submitted by Santiago Horgan, MD, University of Illinois, Chicago.*

Table 3. General: Comparison of Laparoscopic and *da Vinci* Heller Myotomy.



da Vinci OR in action in Munich. Surgical arms and surgeon's console.

working with the tissue.

The InSite Vision System uses the vision portion of the console — the high res 3D endoscope and image processing equipment — to give the doctor real-life 3D images of the area of surgery. Images are optimized by synchronizers, illuminators, and camera controls. The custom dual-lens endoscope is coupled with two three-chip cameras.

Camera control is enabled by a combination of hand controls and foot pedals. The camera can be repositioned or zoom in or out.

The EndoWrist surgical instruments include several forceps, needle drivers, scissors, monopolar and bipolar electrocautery instruments, scalpels, and more. Instruments are available in both 5 and 8 mm diameters.

When an instrument is added to one of the da Vinci system arms, the system recognizes the type and function of the instrument, and displays the number of uses of which it is capable. The da Vinci system detects when you need to use another instrument for what you are doing.

The Proof is in the Results

As the results in Tables 1-3 show (courtesy of Intuitive Surgical, Inc.), the improvements in surgical results with the da Vinci surgical system are amazing. Operative time is lessened, blood loss is drastically decreased, remaining cancer after cancer surgery is minimal, complications are markedly decreased, mortality is down, and health is up. **SV**

RESOURCES

www.intuitivesurgical.com/index.aspx
Intuitive Surgical, creators of the da Vinci Surgical System.

www.intuitivesurgical.com/products/da_vinci_video_overview.aspx
Video and sound overview of the da Vinci Surgical System.

www.intuitivesurgical.com/products/robotic/index.aspx
Video interview with surgeon Mark Talamini, M.D., of the Johns Hopkins University School of Medicine about the da Vinci Surgical Robot.

www.davinciprostatectomy.com/to_dvp.html
The da Vinci prostatectomy.

www.intuitivesurgical.com/corporate/newsroom/events/webcasts.aspx
Live surgery webcast archive.

Other robotic surgery resources:
www.jeffersonhospital.org/webcast/prostate

www.clevelandclinic.org/heartcenter/pub/history/future/robotics.asp

www.whhs.com/services/robotic surgery/overview.htm

www.foresight.org/Nanomedicine/Surgery.html

www.intuitivesurgical.com/corporate/newsroom/videos/media.aspx
Several media videos about da Vinci.

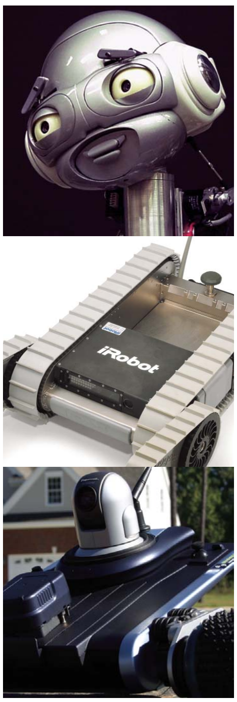
ENDOWRIST FORCEPS

The Resano Forceps offer a Resano style jaw, firm control of tissue, a tip closing jaw, and precise tissue handling. These are suited to handling the peridardium, atrium, and heart valve leaflets.

The endo wrist and the surgeon's hand as it moves to position the instrument.

www.robobusiness2006.com

ROBO Business 2006
CONFERENCE & EXPO



Photos courtesy of Carnegie Mellon University, Openware Robotics, ActivMedia/Mobile Robots and Frontline Robotics.

June 20-21, 2006
Sheraton Station Square
Pittsburgh, PA

The International Business Development Event for the Mobile Robotics and Intelligent Systems Industry

Five Comprehensive Tracks

- Business Development and Partnership
- Technology and Standards
- Applications and Products
- Investment Opportunities
- Markets and Industries
  - Security and Defense Robotics
  - Intelligent Transportation and Field Robotics
  - Healthcare Robotics
  - Consumer Robotics

Additional Sponsors
Advanced Digital Logic
American Robot
Applied Perception
Automatika
Autonomous Solutions
Clean Power Resources
Diversified Case Company
Energid
Evolution Robotics
Foster-Miller
Future Robotics
IZI Robotics
MBus
MobileRobots
Pittsburg Pattern Recognition
RC Driver
RE2
RoboTech Center
ROBOTIS
Segway
ShopBot Tools
Skeyes Unlimited
Systronix
Tobyhanna Army Depot
Valde Systems
Yost Engineering

Association Co-Sponsors
Allegheny Conference on Community Development
Center of Excellence for Remote and Medically Under-Served Areas
Global Emerging Technology Institute
Innovation Works
Mass Technology Leadership Council
MedDev Group Boston Forum
Object Management Group
Pittsburgh Regional Alliance
Robot Hall of Fame
Robotics Society of America
West Virginia High Technology Consortium Foundation

Founding Sponsor: iRobot™

Premier Sponsors: Innova Holdings, Inc. Bridging Innovative Technologies; The Technology Collaborative Accelerating Digital & Robotic Innovation.

Gold Sponsor: CoroWare

Premier Media Sponsor: Popular Science

Produced by: roboticstrends; IDG World Expo

Keynote Luncheon Sponsor: Honeywell

Academic Co-sponsor: Carnegie Mellon

Rubberbands and BAILING WIRE

by Jack Buffington

Keeping In Step

How to Drive a Stepper Motor

Often new robotic hobbyists decide that stepper motors are the best way to move their robot and will start asking how to control them. Often they will also have read somewhere about being able to control stepper motors using the parallel port of their computer.

While stepper motors can be used to drive a robot around or position an actuator arm, and you can — with a little additional circuitry — use a parallel port to control stepper motors, neither of these are optimal for most robotic projects. A computer is better suited to high-level processing and, besides drawing a tremendous amount of power, makes for a very expensive replacement for a microcontroller, such as a BASIC Stamp or a PIC, which is better suited for this task. Stepper motors are great devices in some situations and shouldn't be discounted entirely, but a little knowledge will go a long way towards picking the ideal motor for your robot.

Let's look at what a stepper motor is and how it is constructed. A stepper motor is a special type of motor that moves in discrete steps. It can allow you to precisely position something without the need for an encoder to give you feedback about its position.

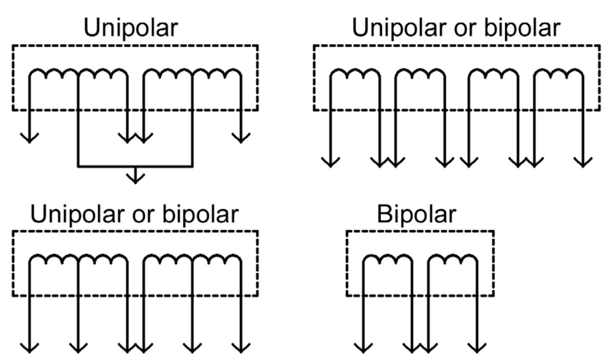
Stepper motors are like permanent-magnet DC motors as far as torque goes. When they are driven at a slow speed, they have plenty of torque. When they are driven at higher speeds, they have less torque. The difference is that while permanent-magnet DC motors have a fairly linear torque curve, the torque for a stepper motor decreases fairly rapidly as the motor speeds up. One additional fact you should be aware of when designing with stepper motors is that they don't take excess torque gracefully. If you pass the rated torque for the speed you are going, the motor will start to lose sync and either fall behind where it is supposed to be or come to a complete stop. This can be a major problem if your robot is likely to encounter a wide range of torque conditions. If you intend to use a stepper motor in your robot, you will need to make sure that you size it so that it can handle any foreseeable amount of torque.

Because of the slipping effect, stepper motors are usually only used in situations where the load on the motor can be predicted ahead of time such as in printers, CNC milling machines, or precise fluid pumping.

Stepper motors come in a wide variety of steps per revolution, but you will find that most stepper motors have 200 or 400 steps per revolution. Stepper motors will have from four to eight wires coming out of them, depending on how they are wound. Usually, they will have four or six wires. Figure 1 shows four different ways that stepper motors can be wired internally. Regardless of the number of wires, all stepper motors operate in a similar manner: you need to energize the coils of the motor in sequence to get it to rotate.

Let's look at Figure 2. In this drawing, you can see a unipolar stepper motor with leads labeled A through D. To the right of the motor are the waveforms that are fed to the motor to get it to turn in the desired direction. The top two patterns use full step patterns. The bottom pattern is what is called half stepping. Half stepping allows you to move the motor through twice as many steps per revolution. There is a third method of driving a stepper motor. This method is called microstepping. Microstepping allows you to position the motor anywhere between the

Figure 1. Four different ways that stepper motors can be wired internally.



NOTE

Last month, I indicated that this month's column would be about using a transceiver from Nordic Semiconductor. In order to provide the readers with the highest quality information, I always make a test circuit and code to ensure that this column is as accurate as possible. There has been some delay with the transceiver circuit, which has prevented it from getting done for this month. Look for information about this circuit in a future column.

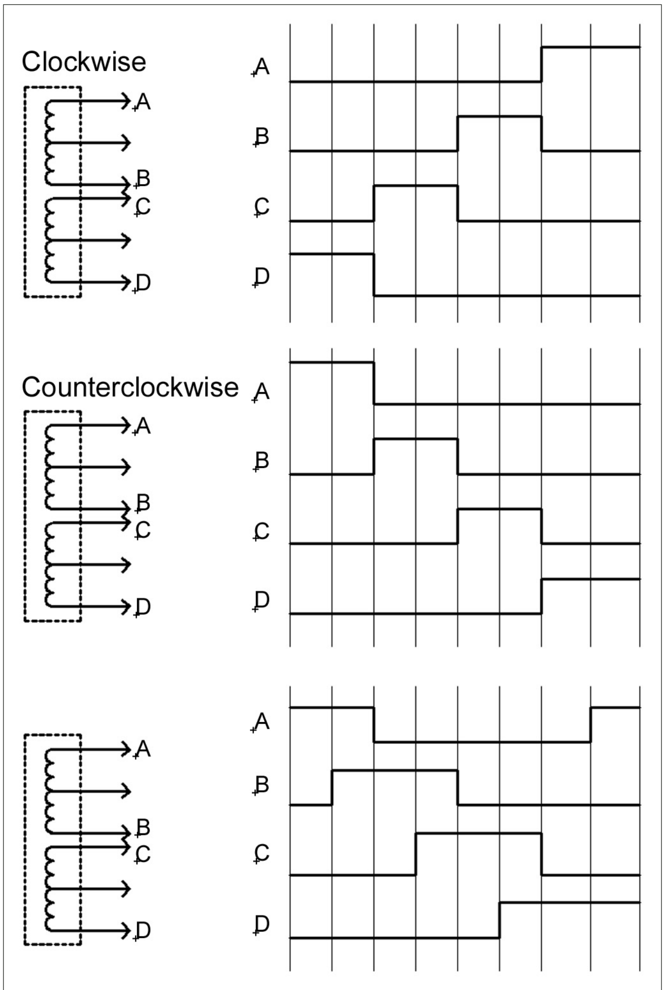


Figure 2. Signals to drive stepper motors.

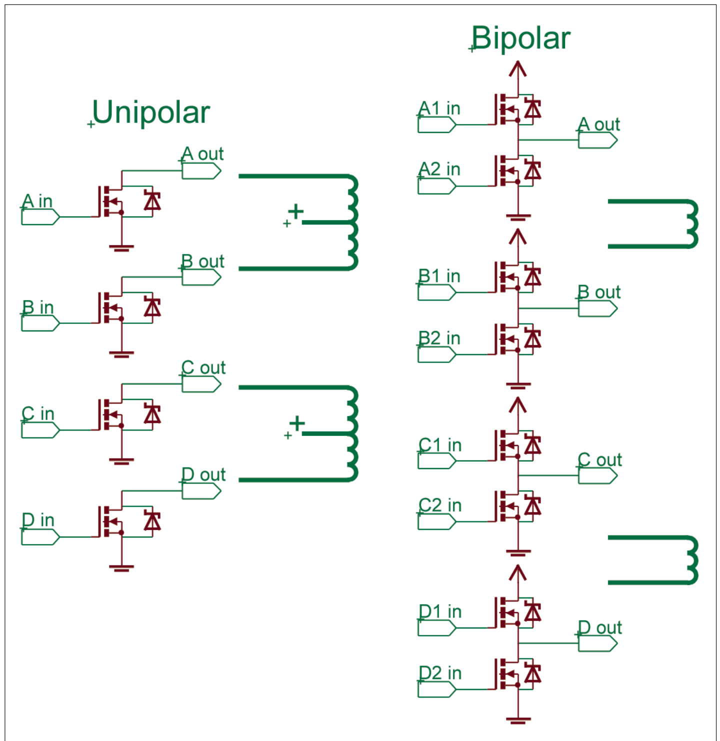


Figure 3. Circuits to drive bipolar and unipolar stepper motors.

whole steps. All you need to do to microstep a stepper motor is to pulse width modulate the leads in a sinusoidal pattern. Be aware, though, that microstepping significantly decreases the amount of torque available.

As you can see, stepper motors are controlled with essentially digital signals. This makes them easy to connect to microcontrollers. Figure 3 shows how to drive unipolar and bipolar stepper motors from a digital circuit. As you can see, the bipolar stepper motor requires two H-bridges to drive the coils in either direction. For the hobbyist, going with a unipolar stepper motor is much easier, though professional applications often use bipolar stepper motors because of their different speed and torque curves.

Let's look now at the software necessary to drive a stepper motor. Without any acceleration, a stepper motor is limited to a small range of speeds. You will need to ramp up the speed of the motor to get it up to its top speed. Most commercially available stepper motor controllers allow you to ramp up the speed of a stepper motor using a trapezoidal speed profile. Let's look at how you can do that using a standard microcontroller.

Ideally, when you are changing the speed of a motor, you would like to make each step a different length. This would give you the smoothest acceleration possible. In practice though, this can be a bit tricky. You could make the duration of each step be some fraction of the previous one. For example, each step could be 254/256ths of the previous step for a slow acceleration or maybe 180/256ths for a fast acceleration. This strategy works but it has a major Achilles heel in that, if you accelerate this way, you aren't guaranteed to have the same number of steps in the acceleration as the deceleration due to the inaccuracy of the math that you are using. Another way would be to calculate through a formula exactly how long each step should take. This would be a great way to do things if you had a lot of processing power available and no other tasks running on the processor.

Unfortunately, in the world of microcontrollers, processing power is somewhat limited, so a third solution to this problem can be used. This strategy uses a lookup table to figure out how long to wait between the pulses. This strategy will allow you to specify how fast the motor accelerates and what the top speed of the motor will be. In this example, Timer 1 on a PIC processor will control the timing of the steps. The clock speed will be 4 MHz, which will divide down internally to one million instructions per second. Timer 1 is a 16-bit timer that counts upwards. When Timer 1 overflows, it will generate an interrupt. If you set the timer to zero every time an interrupt happens, you will get approximately 15 interrupts per second. If you were to set it to 50,000 every time it overflowed, then you would have 64 interrupts per second. To get realistic step rates, you'll need to reset the timer to values around 65,300.

Let's look at how the lookup table that determines the step rate is generated. To get smooth acceleration, your

lookup table will need to have data that graphs out to look like Figure 4. The formula used to arrive at these data points is 65,535 – (range/N). The value of 'range' will determine the slowest speed that the motor can turn. The value used for this column is 5,000. You can figure out what the corresponding step rate would be by dividing your instruction rate by range. In the case of this column, it would be 200 steps per second. This might seem fairly fast, but if you are half stepping your stepper motor and have a 400 step per revolution motor, then it will take four seconds to make a complete revolution.

The next thing that you will need to do is determine the maximum speed that your motor can go. This is limited by two things. The first is how quickly your processor can actually execute its interrupts. In the case of the code presented here, the interrupt can execute in about 70 cycles. This gives approximately 14,300 steps per second. This is a fairly fast step rate. The other limiting factor is how quickly your motor can actually step. Stepper motors max out at a speed that is relatively low compared to permanent-magnet DC motors. Their maximum speed can be as low as 2,500 RPM. The voltage that you are running the motor at will also play a part in determining the maximum step rate of the motor. A good rule of thumb is to start by just using the maximum speed that your processor can handle. You can determine the real maximum speed later if it is lower.

The final variable in the formula is 'N.' This variable transitions from 1.0 to some number that makes the last value in the resulting lookup table be a value that causes the motor to step at its fastest rate. In the case of this column, that value is 65,465 (65,535-70). For this column, the lookup table was generated in Borland C++ Builder, but you could always have your PIC generate the lookup table the first time that it was powered up and store it in its Flash memory. Note that it was chosen to have 128 different speeds that the motor could go. In practice, this is probably more than are needed since you will only notice the speed changes if you accelerate your motor extremely slowly. In most cases you will want to make it spin up to speed as quickly as possible.

Okay! Now you have a nifty lookup table. Let's look at how you can use it. If you were to simply load Timer 1 with each successive lookup table entry each time you stepped, you would be able to get your motor up to speed in 128 steps. In some cases that would be acceptable but why stop there? With a little extra bit of code, you can make the motor accelerate at any rate that you would like. Here is how it is done: You will need to have a 16-bit variable that will help you determine which lookup table value you would like to use. When the motor is stopped, this variable will be zero. Each time the motor steps, you will add an 'acceleration' value to this variable. The upper eight bits of the 16-bit variable will determine which value to use from the lookup table. If your acceleration value is low, it will take a long time to accelerate, if it is high, your motor will accelerate quickly. Decelerating the motor is as simple as taking the same number of steps that you took to accelerate but subtracting the acceleration value from the 16-bit variable each time it steps.

To drive the motor to a new position, you will want to ramp it up in speed, conditionally drive it at a constant rate for a bit, and then ramp it back down to a stop. How many steps will you need to get it up to speed? Let's look at how you can figure that out. First you will need to find how many steps it would take to get to top speed with the current acceleration rate. This is easy enough to do. Simply take how many values are in your lookup table and multiply that number by 256, then divide by your acceleration value. The result will be how many steps it will take you to get to top speed.

If you are limiting your motor's speed to less than its full speed, you will want to reduce the number of steps that it will accelerate and decelerate. To do this, you will use a variable that defines a percentage of the top speed. This value will be from zero to 255. Take the number of steps needed to accelerate to top speed and multiply by your percentage variable. Now divide by 256. This gives you the number of steps it will take to get to the requested speed.

If you just have a short number of steps you would like your motor to make, you will need to limit the number of steps even further. To do this, compare the total number of steps that you will be taking to the number of ramping steps that you have just calculated multiplied by two. You are multiplying by two because you will need to both accelerate and decelerate. If the total number of steps is less than the number of ramping steps, then make the number of your ramping steps be equal to the total number of steps

Figure 4. A graph of a lookup table used to accelerate a stepper motor.

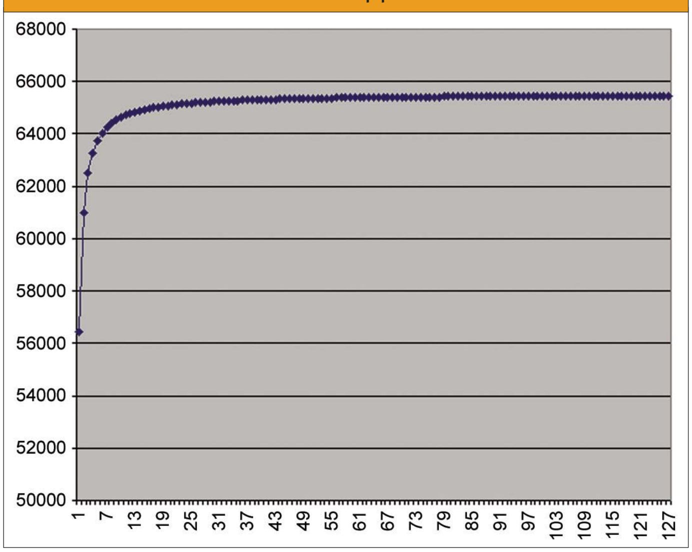


Figure 5. Code to generate a lookup table.

```
const float scaleFactor = .55;
for(int I = 1; I < 128; I ++)
   {
   temp = range;
   Ifloat = I;
   temp /= (Ifloat * scaleFactor);
   temp = 65535 - temp;

   lookup[I] = temp
   }
```

```
int16 calculateRampingSteps(int16 numberOfSteps, int16 accelSpeed, int8 maxSpeedPercent)
   {// Figures out how many steps the motor should accelerate/decelerate for.
   int16 maxPossibleSteps;
   int32 temp32;

   // first figure out how many steps it takes to get to maximum speed ignoring
   // whether that would take the motor past its destination
   // 16384 since only seven bits are used for the lookup table
   maxPossibleSteps = 16384 / accelSpeed;

   // now figure out how many steps it would need to take to get to the top allowed speed
   temp32 = maxPossibleSteps;
   temp32 *= maxSpeedPercent;
   maxPossibleSteps = temp32 / 256;

   // now figure out if the motor can really move that many steps to accelerate
   // and decelerate or if that will be more steps than the move allows.
   if(maxPossibleSteps * 2 > numberOfSteps) // times two because it will decelerate too.
      return numberOfSteps / 2;
   else
      return maxPossibleSteps;
   }
```

Figure 6. Code to figure out how many steps to take for ramping.

divided by two. Take a look at Figure 6 to get a better understanding of how this works. You can find a complete copy of this code on *SERVO*'s website at **www.servomagazine.com**

Finally, you will need to figure out how many steps you will need to take at your top speed. To do this, subtract the number of ramping steps times two from the total number of steps.

You now know how to generate a lookup table, build a circuit to drive stepper motors, and to ramp up and down. The only thing missing is how to tie all of these together. In the example code, you will see that the Timer 1 interrupt is used for the timing of the steps. In the main code, a routine is called that sets up the variables used by the interrupt routine and then enables the Timer 1 interrupt. This routine simply waits until the motor is done making its move and the 'movementDone' variable becomes true before it returns. If your processor has other things to do while moving the motor, you might elect instead to check the 'movementDone' variable before calling the 'moveMotor' subroutine again.

Inside the Timer 1 interrupt, a switch statement is used to determine whether the interrupt should ramp up, run at a constant speed, or ramp down. When the interrupt routine is first enabled, rampSteps, constantSteps, and rampDownSteps have been loaded with how many steps to do in each phase. Each time the interrupt routine is called, rampSteps is decreased by one until it hits zero. At that point, the variable for the switch statement is changed to point execution towards the run-at-constant-speed code. This continues until all of the constant-speed steps have been done. It will then change the switch statement's variable again to point towards the ramp down code.

You are now ready to move your motor! Stepper motors are somewhat complicated to control when compared to permanent-magnet DC motors, but it can be pretty rewarding to see one ramp its speed up and down using code that you wrote. In the right situation, a stepper can provide a much less expensive option for reliable motion control than a permanent-magnet DC servomotor. Your application will determine whether using a stepper motor is a good choice, but if it is, you now have the knowledge to be able to launch into designing for them. **SV**

RESOURCES

Jameco — **www.jameco.com**
Sells stepper motors of various types.

Custom Computer Services, Inc. — **www.ccsinfo.com**
Sells the C compiler that was used with the PIC example code.

Borland — **www.borland.com**
Sells the C compiler that was used with the PC example code.

Microchip — **www.microchip.com**
Manufactures the PIC microcontroller.

AutoFlex Success

by Michael Wittman

Being a new subscriber to *SERVO*, I was impressed with the relevance of the articles and advertisers to the robotics and engineering industries. I am even more impressed because of an article I read in the March 2006 issue of *SERVO*, titled *Autonomous Operation for FIRST Robots* written by Brian Cieslak, a mentor from Team 1675 — The Ultimate Protection Squad.

2006 Aim High

I am a coach and team leader for Team 1714, More Robotics. We are a rookie team this season competing in the 2006 FIRST Robotics Competition called Aim High. Aim High challenges two alliances, one red and one blue, composed of three teams each, to attain a higher score than their opponent. They do this by creating a robot that can project a seven-inch diameter foam ball through a 30-inch diameter round target (the high goal) that is 8.5 ft above the ground, or project that same foam ball through a low goal that measures 10" x 24".

Stalled at the Starting Line

Like most rookie teams, we made all the classic rookie mistakes that first-year teams make. We arrived at our first regional competition with mechanical issues yet to be solved, a slightly overweight robot, and we aimed a little too high for what we could accomplish with our programming. Well, at least, this is what we thought. In addition to all of that, we decided not to have an autonomous program because we

The first AutoFlex test run.

just didn't have the time to work it out.

What we found was that many teams, not just the rookies, had similar issues. After a preliminary inspection of our robot, the Inspection Officials confirmed that we were indeed overweight by 1.4 lb and that we were oversized by about 1/8 inch, in both width and depth. What they couldn't tell us was that we had programming issues, also.

A FIRST competition occurs over three days. The first day, usually a Thursday, is for uncrating your robot, passing robot inspection, fixing issues, and running practice sessions. After our preliminary inspection, we confidently placed our robot on the game field ready for our first practice action. I should tell you at this point that we had no reason to think our robot wouldn't function because the night before we crated it, it drove and handled fantastically. Actually it was the only thing we were confident about.

The Aim High game starts with a 10-second autonomous period, so when the game started, we did not expect our robot to move. When the autonomous period ended we were surprised to find that our robot moved forward about one inch, stopped, and wouldn't do anything. The electronics official from Innovation First who was there to monitor such things told us our robot was actually doing something; it just wouldn't move. He said something about an infinite loop and our program code needed to be changed.

Did I mention we were a rookie team? Did I mention that our programming coach was not able to come to our regional competition? Did I mention that we didn't have a clue where the problem in our code was or how to correct it?

AutoFlex to the Rescue

Nate setting up the AutoFlex to record.

Sarah, our team captain, said that she had read an article in *SERVO* about the program AutoFlex and how it was used with Team 1675's robot for autonomous programming. This team happened to be across the aisle from us in the pits. Sarah suggested that our programmers, Nate and Joey, talk with the programmers from Team 1675 about getting help to fix the program. Here is where our new best friends came into play — Team 1675, The Ultimate Protection Squad. Specifically, Brian Cieslak, a mentor for Team 1675, and Andrew, a student on that team, came over to our pit area. With the gracious professionalism FIRST teams show, they offered to help us out.

Nate, Andrew, and Brian went off with our programming laptop to look at the code. After about an hour of deliberation, Nate came back and said he wanted to completely change how our controls work and how the robot drives. Sarah and her teammates looked at Nate and asked if he was crazy. "It took us six weeks to get it to do what it does now," said Dustin, a human player on our team.

Nate said Brian and Andrew were going to help, and that we could have it up and running in about 60 minutes. We all agreed that it couldn't operate any worse than it does now, so we gave him our blessing and prayed that it would work okay. About 45 minutes later, Nate and Andrew came back and confidently said they were ready.

Nate said, "Oh, by the way, do you think we should try to do autonomous mode also?" Here is where AutoFlex enters the story, but I will finish the reprogramming story first. Nate, with the help of Joey, and with Andrew overseeing the whole scene, reprogrammed the team robot. We brought it to the practice area and had our team driver, Matt, take it for a spin.

Matt said, "It's great, certainly better than it was before, just sitting there when I moved the controls."

I said, "Okay Nate, tell us about what you can do with autonomous mode and how long will it take."

Nate explained that AutoFlex, the name Team 1675 gave to the program, records the actions that the robot driver creates when he drives the robot. After those actions were compiled into a file and installed onto our robot controller, we would have an autonomous mode that matched

exactly what our driver did.

"Our robot will do anything our robot driver can make it do," Nate said.

So I said, "Great Nate, how many weeks do we need to make this work?"

Nate laughed and said that he and Andrew had already worked out the details and that they would need about 30 minutes total time. I was impressed.

Sarah asked, "What are we waiting for?" Matt said, "It will be cool to have autonomous. Let's do it."

The day was getting late. We had already run our last practice match, so we decided to measure off a test run, record it, and see what happened. Sure enough, with Brian and Andrew to guide us, Nate set up the program-record feature, and Matt drove the robot. In less than 15 minutes, we had completed a 10-second autonomous program, downloaded it to the robot, and played it back. We were all getting excited, and the day was coming to an end, so we decided to record our competition autonomous program first thing in the morning when we came in.

Going for the Goal

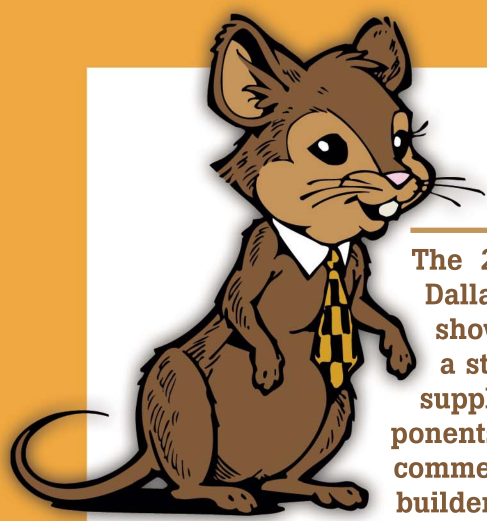
The next morning we were scheduled to run our first qualification round in Match 5, so we didn't have a lot of time. The night before we agreed that we would go forward about eight feet, turn right, and shoot two balls at the goal. Robots were allowed to start with 10 balls, but we did not want to spend all of them in case we missed or needed them at a more strategic time during game play. We set up the run and were happy with the drive the second time through, so we downloaded it to the robot. We had just heard our second call to queue up for our match so we had to go to the field without testing it to see if it worked. We were apprehensive to say the least.

Matt driving the robot while recording.

Our competition team, Sarah, Matt, Nate, and Dustin put the robot on the field when our turn came and all I can say is WOW. It worked perfectly. We missed the target just a little to the left, but the balls hit the backboard at just the right height. We decided not to change anything except lining the robot up differently.

Our next qualification round was Match 10. The same competition team set up the robot so it was adjusted slightly differently from the previous match. When the match started, our robot came out and turned perfectly. It shot off two balls that went dead center through the upper target. Our team in the stands went wild. Making those shots helped our alliance win the match.

Throughout the day, we ran the same program in autonomous mode with fairly consistent results. Our first tournament was a huge success, far beyond what we thought we were capable of doing. Using AutoFlex was far easier than we thought it would be and took less time than writing code from scratch. Having Brian Cieslak and Andrew from Team 1675 help us out was a large part of our success that day. We ended the day winning the Rookie All-Star Award, and left the event with two more heroes. Had it not been for Brian writing that article, or *SERVO* printing it, or Sarah reading it, or Team 1675 helping us, our day would have been significantly different. **SV**



MICROMOUSE

by R. Steven Rainwater

The 2006 Applied Power Electronics Conference was held recently in Dallas, TX at the Hyatt Regency hotel. That's the building that always shows up in photos of the Dallas skyline looking like a geodesic ball on a stick. The APEC conference consists of engineers talking about power supplies, DC-DC power converters, and the latest obscure electronic components. There are presentations, panel discussions, technical papers, and commercial exhibits. APEC also includes a special attraction for robot builders around the world: the annual APEC MICROMOUSE CONTEST.

IEEE Spectrum Magazine announced the first Micromouse contest in their May 1977 issue. It was held two years later in New York. Thousands entered but only 19 competitors actually showed up and ran on the day of the contest. One year later, in 1980, a Micromouse contest was held in conjunction with the Euromicro conference in London. After that, Micromouse competitions spread all over the world.

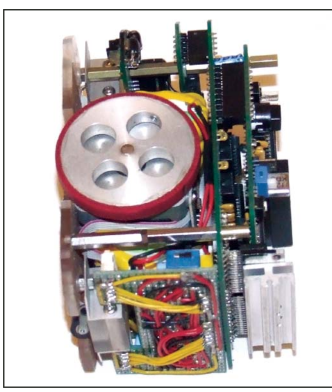
So what is a Micromouse and how does the contest work? As the name implies, a Micromouse is a small robot, no larger than 25 cm x 25 cm. There is no height limit. The course is a 16 x 16 grid of 180 mm x 180 mm squares (for the metric-impaired, that totals to about 10 x 10 feet). A maze is constructed by setting up 50 mm high walls along the grid. The maze paths which the robots travel down are 160 mm wide. Additional rules detail tolerances and other characteristics.

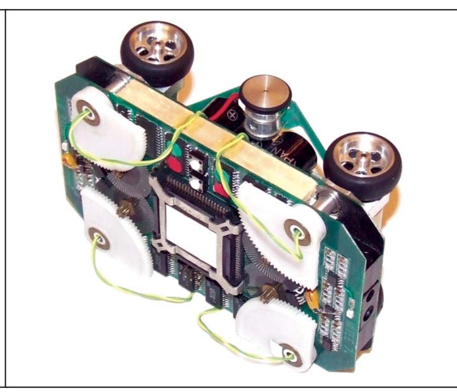
A Micromouse must find its way from a starting point at an outside corner to the center square in the fastest possible time. The key word is fast; very fast. A competitive Micromouse probably moves faster than any robot you've seen at a hobby robotics competition. To win a Micromouse contest 20 years ago, robots could move at a leisurely pace, finding the center of the maze in around five minutes. To have a shot at winning today, your Micromouse had better be able to run the maze in 10 to 20 seconds. It helps that the mouse can make multiple runs, remembering what it learned on previous ones. Each run consists of a "learn phase" in which the mouse searches for paths to the center and a "run phase" in which it takes the fastest path to the center. The final score comes from the run phase time plus a portion of the search phase time. If a mouse avoids crashing or manual restarts, a small bonus may apply.

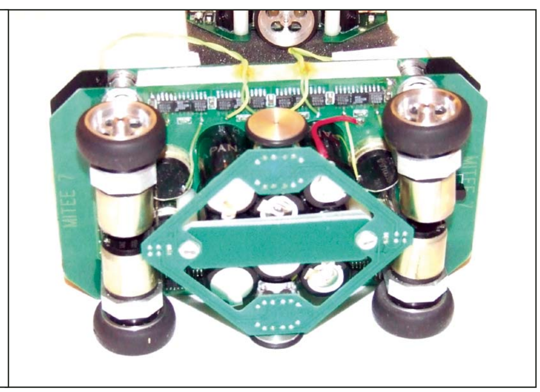
Mouse builders have made constant technological advances allowing the robots to move faster and make sudden, right-angle turns of the sort that inspire UFO reports. A Micromouse needs very good brakes for quick deceleration and powerful motors for equally fast acceleration. Robots that accelerate at 1G and have speeds of five meters/second are not unknown, though they often require exotic mechanisms such as ducted fans to create suction, preventing the Micromouse from flying off the surface.

This obsession with speed has resulted in a strange ritual; the use of lint rollers on the course and robot wheels to remove even the tiniest speck of dust that might cause a speeding robot to lose control. Before the contest begins, special lint rollers on long sticks are rolled over the entire course. Individual contestants bring their own lint rollers, which they use on their robot's wheels before each run and on the course when needed.

Dust isn't the only worry of a Micromouse builder. Slightly out of round or unbalanced wheels can cause a crash. Even maintaining a consistent distance from the course walls becomes difficult at high speeds.



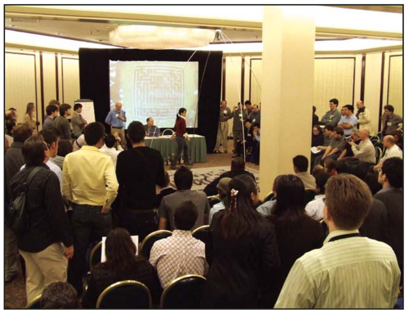




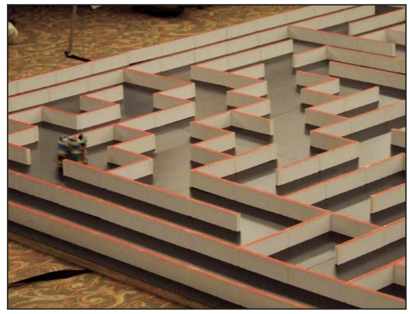
CONTEST 2006 >>

>> Contestants, Start Your Lint Rollers!

>> Raptor of Nanyang Polytechnic making its first run.



>> By the fourth mouse, there was quite a crowd.



>> EXCEL-2 from the Institute of Technical Education in Singapore.

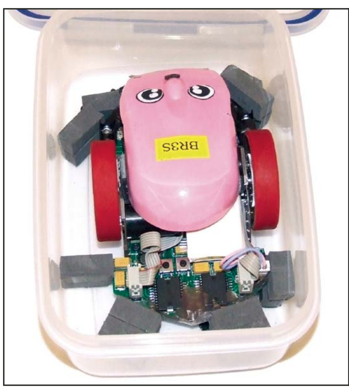
With all this in mind, I arrived at the APEC 2006 contest. Since it was being held in Dallas, I was not surprised to see quite a few members of the Dallas Personal Robotics Group present to view the event. By the time the contest was underway, around one hundred people had crowded into the contest area. A video camera suspended over the course provided an aerial view that was projected onto a large screen for the spectators.

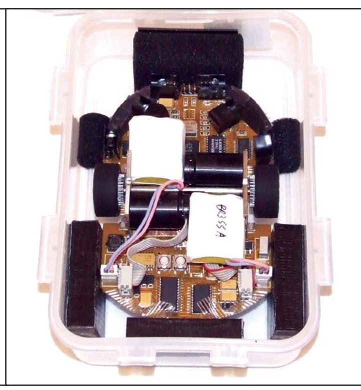
The majority of contestants were from Singapore, with several US entries, as well. Nanyang Polytechnic in Singapore brought several robots with low-slung, titanium frames. One of their robots — BR3S — utilized on-board gyros to provide more accuracy for sharp turns. BR3S, which was decked out with a pink shell complete with nose and eyes, won the prestigious All Japan Micromouse competition in 2005. There were also two robots from the Institute of Technical Education in Singapore. A Vanderbilt University team ran a Micromouse named Ichimokusan (Japanese for "full speed"). Another US entry was MITEE Mouse 9 built by David Otten of MIT. Gary Vigen displayed and described his Mouseketeer robot, which was under construction.

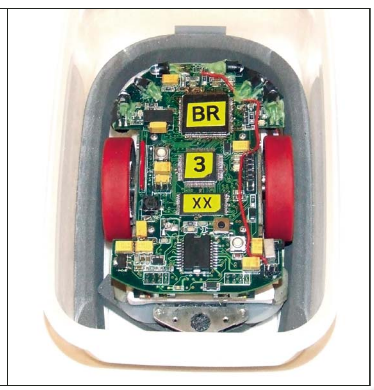
The contest was a blur of speeding mice and spinning lint rollers. In the Open Category, Nanyang Polytechnic's BR4 came in first with a time of 7.48 seconds, beating MITEE Mouse 9 (21.75 seconds) and Dover-2 (23.61 seconds). In the Student Category, the winner was Excel-2 of the Institute of Technical Education with a time of 20.57 seconds. While pink BR3S didn't come in first, it did take the honor of the fastest recorded run during the contest with a time of just 7.36 seconds.

Many of the spectators had never seen a Micromouse contest and were visibly impressed by the speed of the robots. Afterwards, several of us adjourned to the Hyatt's rotating Dome Lounge and talked of speedy robots over drinks. Several inspired DPRG members were considering the idea of building a Micromouse course of their own or perhaps building a Micromouse to enter in an upcoming Micromouse competition.

If you'd like to see a Micromouse competition, there will be several more this year. The UK National Micromouse Competition is set for June 10 at the Technology Innovations Centre in Birmingham. The Singapore Inter-School Micromouse Competition is coming up this summer in Singapore, though a final date has not been set yet. The date for the 27th annual All Japan Micromouse Contest is not finalized either but it will likely be in November. The APEC 2007 Micromouse Contest will be held February 26 at the Disneyland Hotel in Anaheim, CA. SV







>> Presenting: THE MICROMICE! You won't find these mice caught in a trap anytime soon!

EVENTS CALENDAR

Send updates, new listings, corrections, complaints, and suggestions to: **steve@ncc.com** or FAX **972-404-0269**

May and June bring another large assortment of robot competitions. 2006 is shaping up to be one of the busiest years I've seen. I expect the pace to slow down a bit by July, but there are many more events yet to come.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

— R. Steven Rainwater

May

6 **CIRC Central Illinois Bot Brawl**
Peoria, IL
Includes several classes of autonomous sumo and remote-control vehicle destruction.
www.circ.mtco.com/competitions.htm

6 **LVBots Challenge**
Advanced Technologies Academy High School, Las Vegas, NV
Line following, line maze solving, and mini sumo, all for autonomous robots.
www.lvbots.org

9-11 **DTU RoboCup**
Technical University of Denmark, Copenhagen, Denmark
Line following and wall following.
www.robocup.dtu.dk/

10 **Micro-Rato**
University of Aveiro, Aveiro, Portugal
Micro-Rat competition (similar to micromouse, but larger).
http://microrato.ua.pt/

10-11 **Haifa Robot Competition**
Haifa, Israel
Autonomous robot events for university students, high school students, and amateurs.
http://math.haifa.ac.il/robotics/competition/

13 **Atlanta Robot Rally**
Southern Polytechnic, Marietta, GA
Open Contest — contestants choose their own goal for their robot. Vacuum Contest — autonomous household vacuuming contest/mini sumo.
www.botlanta.org/Rally/

13 **DPRG Robot Talent Show**
Dallas, TX
No rules, just talent. If your robot does anything interesting like avoid obstacles, move under its own power, or burst in to colorful flames unexpectedly, you could be a winner. The DPRG has held robot talent shows off and on since the 1980s.
www.dprg.org/competitions/

13 **RoboFest**
Lawrence Technological University, Southfield, MI
Events for LEGO robots and Advanced robots.
http://robofest.net/

13 **Western Canadian Robot Games**
Alberta, Canada
Robot sumo (western rules), mini sumo, walking robot triathlon, robot art contest, Atomic Hockey, and a full set of BEAM events.
www.robotgames.net/robot_games.htm

19 **SPURT (School Projects Using Robot Techniques)**
Rostock-Warnemunde, Germany
Robots race on the official SPURT track.
http://spurt.uni-rostock.de/

19-20 **Micro Air Vehicle Competition**
Brigham Young University, Provo, UT
Surveillance and endurance events for MAVs and an ornithopter competition and design competition.
www.et.byu.edu/groups/wwwmav/Tenth_MAV_Site/

20 **KCRS Robot Exhibition and Competition**
University of Missouri, Kansas City, MO
Line following, mini sumo, and the interestingly named Dinnerware Demolition Derby.
www.kansascityrobotics.org/

20-21 **Mechwars**
Eagan Civic Arena, Eagan, MN
Radio-control vehicles will destroy each other in Minnesota.
www.tcmechwars.com

26 **NATCAR**
UC Davis Campus, Davis, CA

Very high-speed autonomous line following.
www.ece.ucdavis.edu/natcar/

June

1-3 **ION Autonomous Lawnmower Competition**
Dayton, OH
Robots are judged on speed and accuracy.
www.automow.com/

2-4 **Eurobot**
Catania, Italy
Involves black balls, white balls, and holes — robots match off two at a time.
www.eurobot.org/

10 **UK National Micromouse Competition**
Technology Innovations Center, Birmingham, UK
Micromouse builders compete for the coveted Brass Cheese. Try to make it to this Micromouse event. These are amazing little robots.
www.tic.ac.uk/micromouse

10 **Vancouver Robotic Games**
BCIT Campus, Burnaby, British Columbia, Canada
Line following, advanced line following, BEAM Solaroller, BEAM Photovore, and a walker triathlon.
http://vancouverroboticsclub.org/

10-12 **AUVS International Ground Robotics Competition**
Harrison Township, MI
University-built autonomous ground robots navigate an outdoor obstacle course.
www.igvc.org/deploy/

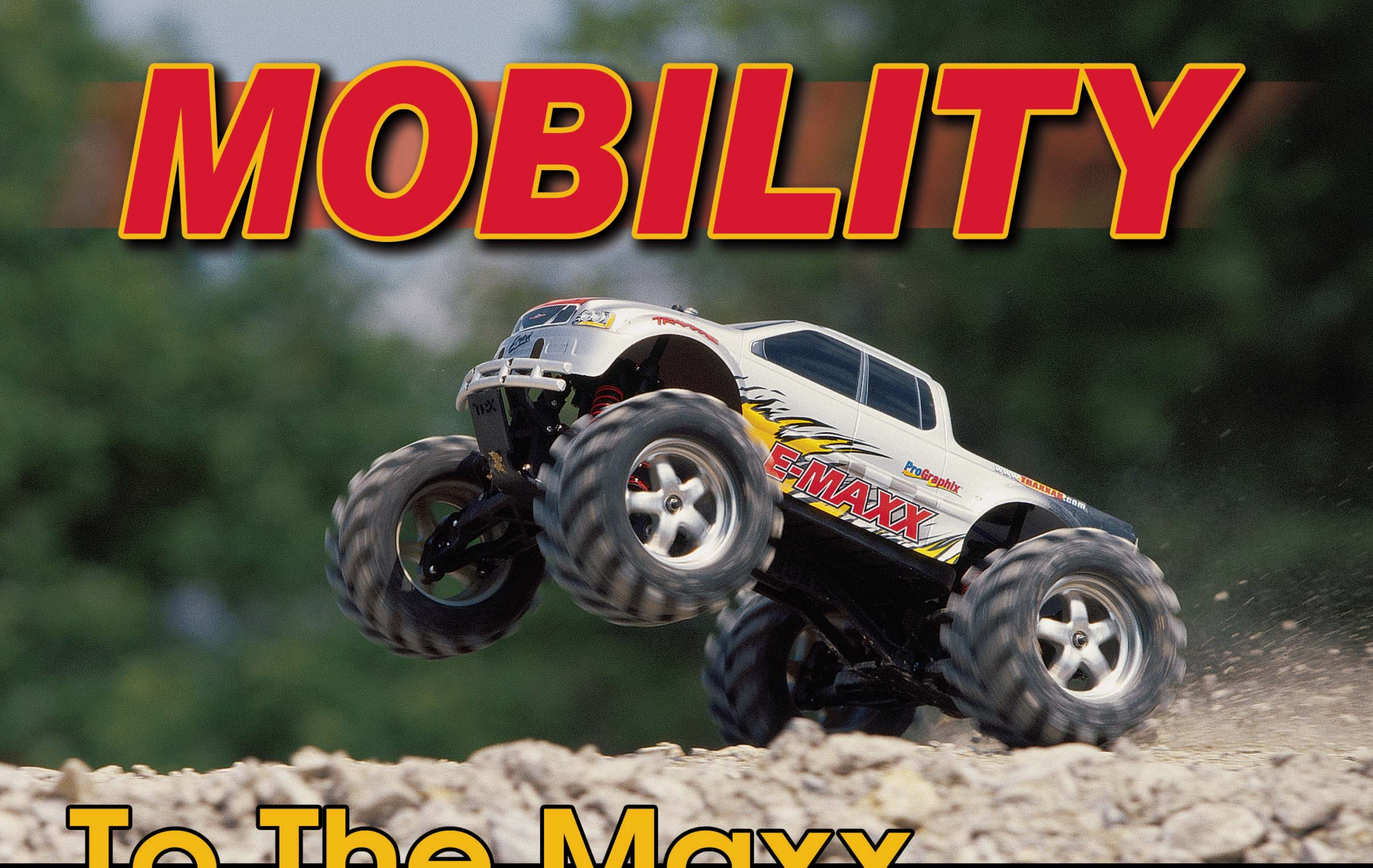
14-20 **RoboCup Robot Soccer World Cup**
Bremen, Germany
All the usual Soccer events: small, mid, humanoid, and AIBO. Also a NIST rescue robot contest. In addition to these events, the Robot Cup@Home competition will be held in conjunction with the World Cup this year.
www.robocup.org/

16-18 **RoboGames**
San Francisco State University, San Francisco, CA
The robot competition formerly known as Robolympics is back. It will have more robots and events than I can list here. Visit the website.
www.robogames.net/

23-25 **MATE ROV Competition**
Neutral Buoyancy Lab, NASA Johnson Space Center, Houston, TX
Student-built ROVs must locate and retrieve objects of varying size and shape.
www.marinetech.org/rov_competition/

24 **Mobile Robotics Software Challenge**
Portland, OR
Computer and software rides a provided robot and reports on path. Also known as the egnellahC dnarG APRAD because it's like the DARPA Grand Challenge in reverse.
www.mobilerobot.org/MRSC.htm

MOBILITY



To The Maxx

PART 2 — Controlling the E-Maxx With Your Computer and a Joystick

by Chris Cooper

Last month, we modified the E-Maxx to make it more suitable as a robotic platform. We improved the steering, increased the torque, strengthened the suspension, and installed a deck for mounting sensors and electronics. Now it's time to take control — ditching the original remote control in favor of a PC and an off-the-shelf USB joystick.

We'll create a robotic control system to control the DC motors, steering, and shifting servos wirelessly via Bluetooth.

To create a robot that is easily upgradeable, we will implement a distributed-control architecture to spread control across multiple processing nodes. That way, we can eventually add odometry, auto-shifting, GPS, digital compass, range sensors, and more to achieve autonomous or semi-autonomous control. With our distributed-control architecture, we can incrementally build the robot and experiment with various approaches to see what works best.

This article shows you how to modify the E-Maxx, including:

- Building a Servo Module
- Building a Wireless Communications Module
- Choosing and Programming an Application Host

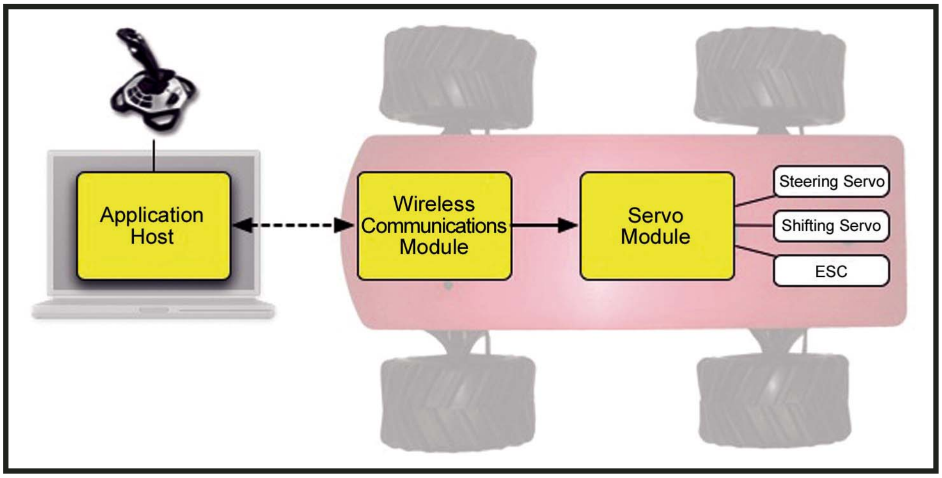
Figure 1 illustrates the relationship of the three modules and their inputs and outputs.

Building the Servo Control Module

For my E-Maxx, I chose a PEC-110 eight-bit Port Extender to control the steering and shifting servos and the Electronic Speed Controller (ESC). This off-the-shelf port extender is designed to support distributed-control architectures and is perfect for controlling hobby servos. Additionally, its firmware

Photo Above: The E-Maxx RC monster truck makes an excellent robotics base. *Photo courtesy of Traxxas.*

Figure 1. Overview of modifications to control the E-Maxx with a Joystick.



eliminates the need to do any embedded programming, which saves a lot of time and effort. If you want to create your own port extender, you can find schematics for one at **www.machinebus.com**

The schematic in Figure 2 shows the Servo Control Module. Pins P7, P8, and P9 are the servo-control signals. The lower grouping of four pins is the serial networking bus.

The breadboard diagram in Figure 3 shows the PEC-110 configured to support three servos. Although not shown in the diagram, the steering servo (CON 2) and the shifting servo (CON1) receive power from the ESC when it's connected to CON3.

Building the Wireless Communications Module

For the Communications Module, I chose an MCI100 UART Interface, which is a low-power, high-performance processor that communicates asynchronously through its Universal Asynchronous Receiver-Transmitter (UART) serial interface. The UART interface works well with the port extender in the Servo Control Module. The two devices communicate via a Controller Area Network (CAN). It worked right out of the box. I use the MCI-100 in the Communications Module to interface the Servo Control Module with the Bluetooth Serial Module.

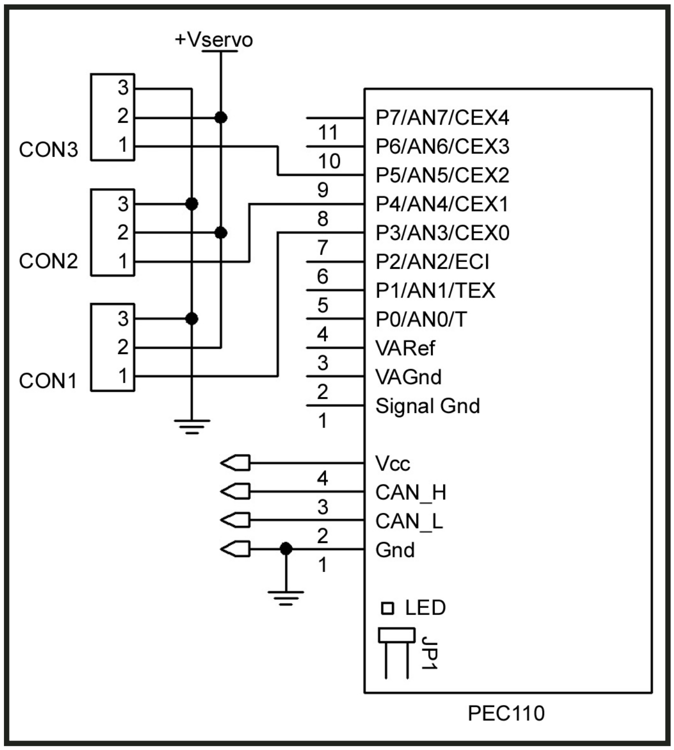
For wireless inter-machine communication with the Host Module, I chose to network my functional modules using BlueSMiRF Extended Bluetooth Serial Module from Spark Fun Electronics. Bluetooth is used for our wireless communication because of its low cost and ease of use. If you prefer, you can use WiFi or serial RF instead.

Bluetooth is a radio standard, primarily designed for low power, short-range communication. Using the Bluetooth Serial Device Profile, the wireless connection is treated just like another serial port.

The RF-BlueSMiRF Bluetooth-to-serial module connects the MCI-100 to the host. The extended version has hardware flow control and is a Class 1 Bluetooth device so it has a range of approximately 100 m.

Figure 4 shows the schematic for the Communication

Figure 2. A schematic of the Servo Control Module.



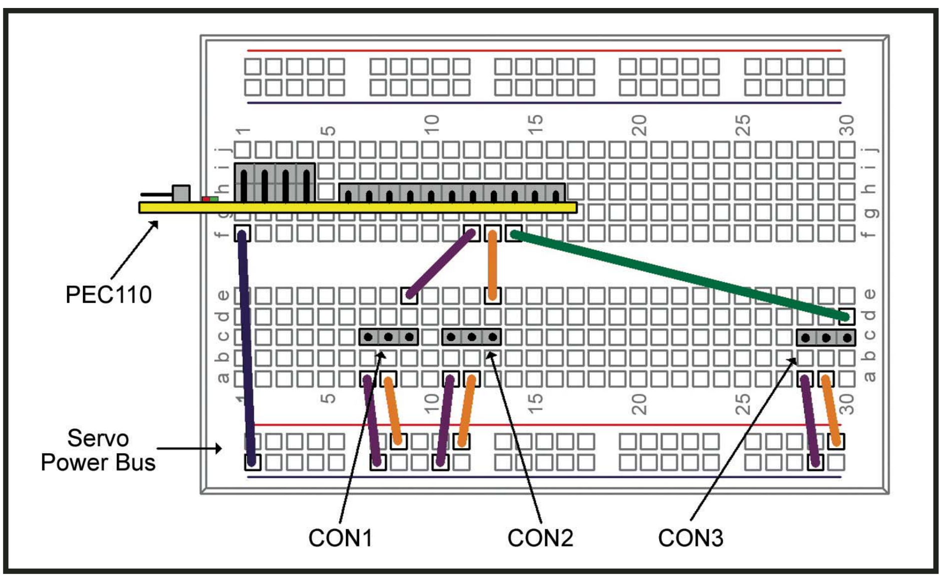


Figure 3. The Servo Control Module outputs proportional width pulses to each servo and to the Electronic Speed Controller (ESC).

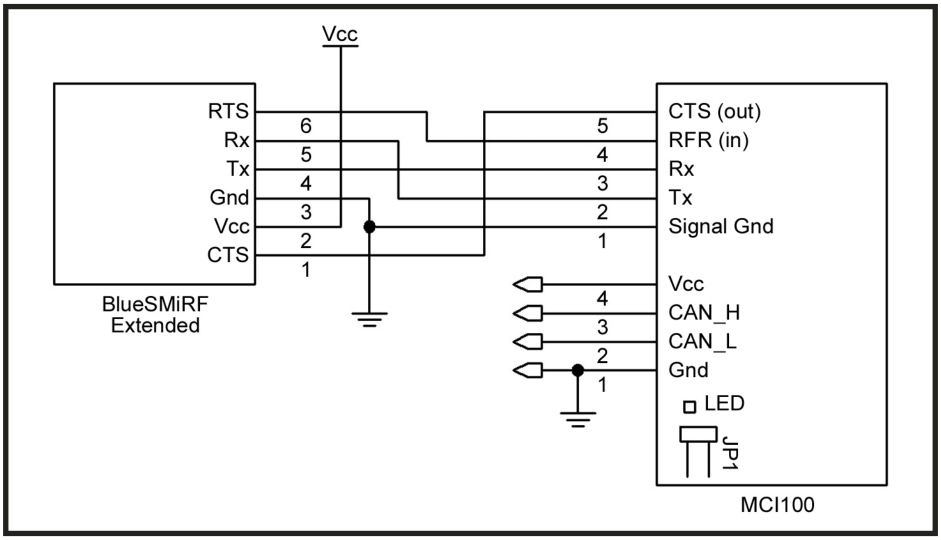


Figure 4. The schematic for the Communication Module.

Module. The breadboard diagram in Figure 5 shows how simple it is to connect the MCI-100 and the BlueSMiRF.

Choosing and Programming the Application Host

By using a distributed-control architecture, we could have chosen any number of platforms for a host, including a PDA or palm computer. I chose a Personal Computer (PC) because it is familiar, provides a rich environment for developing applications, and it easily connects to Bluetooth and a joystick. Using a PC for the host provides a platform capable of processor and memory-intensive algorithms, as well as complex visual displays. It's also a great environment for iterative development, testing, and debugging.

One of the nice things about the chosen hardware is that it doesn't require embedded programming. The only software written was on the host. In fact, I am able to treat the servos connected to the servo control module as if they were connected directly to the host PC, making my code/debug/test cycle very fast.

Next, to program the application host, we must receive input from the joystick, map joystick events into servo positions, and send instructions to the servos.

Receiving Input From a Joystick

While it is now possible to drive the E-Maxx from the host module, it makes more sense to drive it using a joystick. The SDL project — a free cross-platform multi-media development API written in C — has excellent support for USB joysticks, as well as OpenGL, video, audio, threading, timers, and endian independence.

The SDL_Joystick implementation supports both manual inquiry and event processing. In event mode, joystick events are stored in a queue to be processed by the application when possible. The benefit of event mode is that all events are stored in a queue, so no event is missed. The downside is that if events are added to the queue faster than they can be processed by the application, the next event in the queue may not represent the current state of the joystick. In manual mode, the application queries and holds the current state of the joystick, which can cause joystick actions to be missed if the state is not updated often enough. After trying both methods, I found there was a perceivable lag between joystick action and servo response in event mode, but manual mode was very responsive.

There are essentially two joystick actions that need to be processed: button pressed and axis movement.

`SDL_JoystickGetButton()` returns a "1" if the specified button was pressed when the last `SDL_JoystickUpdate()` was called.

`SDL_JoystickGetAxis()` returns the position of the specified axis the last time `SDL_JoystickUpdate()` was called. The position of the axis falls within the range -32768 to 32767.

BENEFITS OF USING A DISTRIBUTED CONTROL ARCHITECTURE

Modularity

⇛ Manage complexity by decomposing into smaller, more logical units.

Reusability

⇛ Reusable hardware and software modules make prototyping fast, easy, and inexpensive.

⇛ Modules with differing capabilities can be combined to fit any application.

Scalability

⇛ Processing nodes can have various speed, memory, and I/O capabilities.

⇛ Processing nodes can have varied functional responsibilities.

Extensibility

⇛ Adding new functionality is as easy as adding modules and testing them.

⇛ Integration of many different devices and communication protocols is possible.

⇛ Modules that provide diagnostic capabilities can be temporarily added to the system.

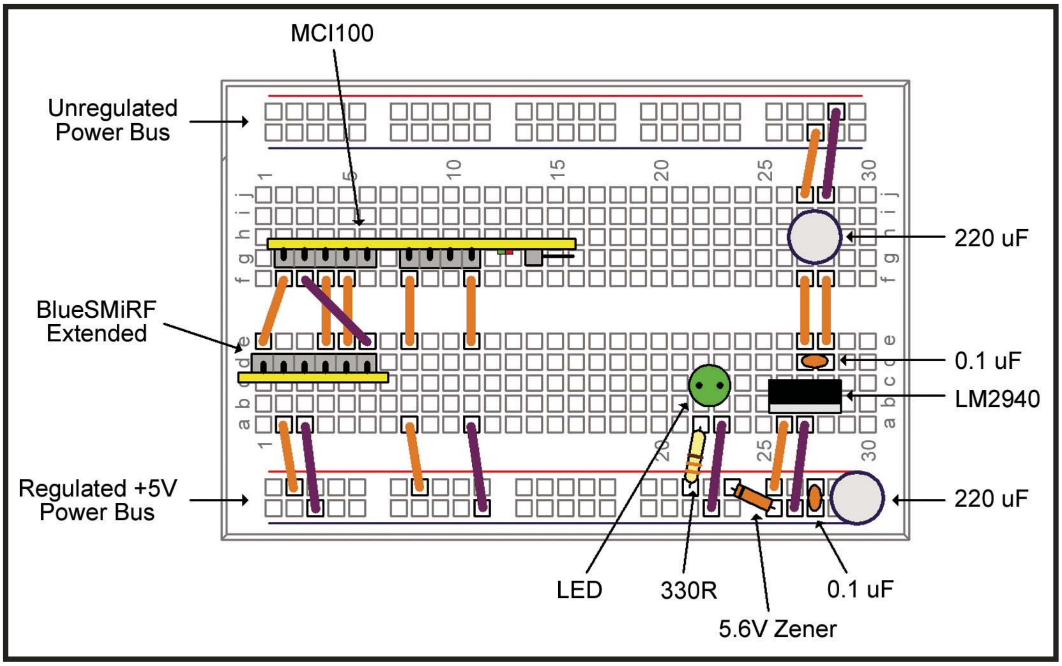
Reliability

⇛ Redundant modules can be used to provide backup or crosschecking.

⇛ Partial failures can be handled gracefully allowing general operation to continue.

(For more information on CAN technology, see "Overview of Controller Area Network" in the Resources.)

Figure 5. The Communication Module bridges Bluetooth to CAN.



Controlling the Servos

Controlling servos from the Host Module is accomplished using the PEC-110 and its Servo API. The PEC-110 can control up to five servos when programmed with the available servo firmware. The Machine Bus Servo API is used to create proxy servos in the Host Module, which then forwards commands issued off to the PEC-110 to execute them.

Controlling the servos is accomplished easily using an abstract data type and some interface methods defined in the servo header file. The interface allows getting and setting the minimum, maximum, current position, and neutral position (the position to return to when the joystick is at center) for each servo. All that's needed to create a servo is a reference to the communications bus and the Servo ID, which corresponds to the number of the three-pin connector for the servo.

Test your servo to establish the minimum and maximum rotation, as well as the neutral position. This is the only way to make sure you're not overdriving it. Trying to send a servo farther than it is supposed to go in any direction will cause jittering, above average current consumption, and will drain your batteries quickly. The easiest way to determine servo parameters is to set a low minimum and a high maximum and a starting position somewhere in between. Slowly increment the position and record the new maximum value just before jitter occurs. Similarly, slowly decrement the position and record the new minimum value just before jitter.

Controlling the Servos With the Joystick

The most intuitive mapping of joystick event to E-Maxx action is to have the stick's X-axis control steering and its Y-axis control forward and reverse. You can control shifting gears using any of the available buttons, so I chose the trigger. Pressing and holding the trigger shifts the E-Maxx into second gear. Releasing it downshifts the E-Maxx back into first gear. In order to "trim" the neutral position while running, two arbitrary buttons are used to increment/decrement the neutral position value. Trimming the steering servo to the proper neutral position allows it to

POWER DISTRIBUTION

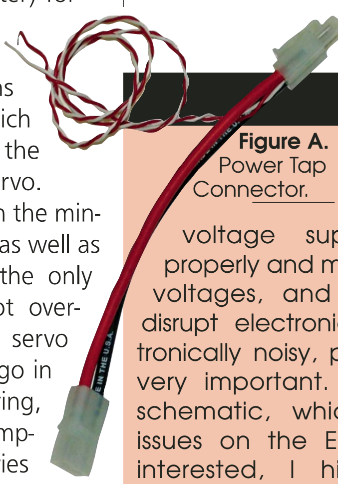


Figure A. Power Tap Connector.

Since digital electronics require a regulated, clean voltage supply to function properly and motors require higher voltages, and currents tend to disrupt electronics by being electronically noisy, power distribution is very important. I've included my schematic, which resolves these issues on the E-Maxx. If you are interested, I highly recommend *Intermediate Robot Building*, by David Cook, for a more in-depth discussion of power regulation.

Rather than pay the weight penalty of using a third battery, I decided to power the digital electronics off one of the existing E-Maxx batteries, with the connector in Figure A.

I can always switch to using a third battery if it becomes necessary.

Figure B shows the power regulation circuit, which includes a zener diode for overvoltage protection. The LM2940 has built-in protection for reverse polarity.

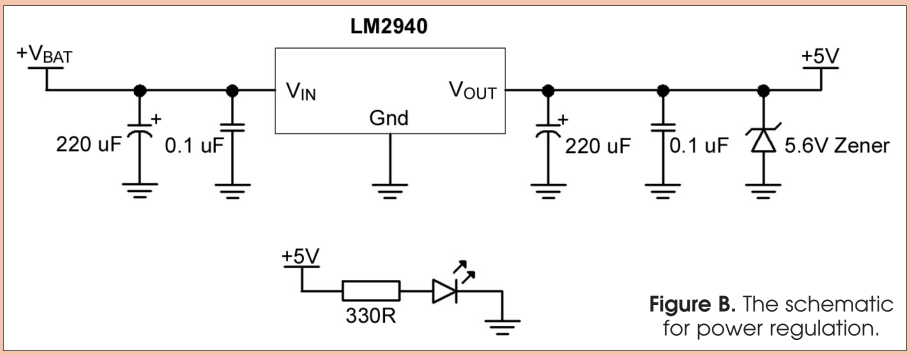


Figure B. The schematic for power regulation.

PARTS LIST

Power
- ⇛ DPDT Switch
- ⇛ 33 Ω Resistors (1 W or more), Digi-Key #P33W-1BK
- ⇛ Green LED and two 1 kΩ Resistors
- ⇛ Zener Diode 1N5232B, Digi-Key #1N5232BTRCT
- ⇛ LM2940CT-50, Digi-Key #LM2940CT50-ND
- ⇛ TO-220 Heatsink, Digi-Key 294-1011-ND
- ⇛ 0.1 µF Capacitors, Digi-Key #BC1127CT (100)
- ⇛ 220 µF 35WV (unregulated bulk) Digi-Key #P5166-ND
- ⇛ 220 µF 10WV (regulated), Digi-Key #P964-ND

Servo Control Module
- ⇛ PEC110 eight-bit Port Extender from Machine Bus (**www.machineBus.com**)

Communication Module
- ⇛ BlueSMiRF Bluetooth Serial Module from Spark Fun Electronics (**www.sparkfun.com**)
- ⇛ MCI-100 UART Interface from Machine Bus

Host Module
- ⇛ Bluetooth USB Dongle from Spark Fun Electronics
- ⇛ Logitech Extreme 3D Pro (**www.logitech.com**)

RESOURCES

⇛ *Intermediate Robot Building*, by David Cook.

⇛ *An Overview of Controller Area Network (CAN) Technology*: **http://machinebus.com/documents/whitePapers/canOverview/canTechnologyOverview.pdf.**

⇛ SDL — Simple Directmedia Layer (**www.libsdl.org**).

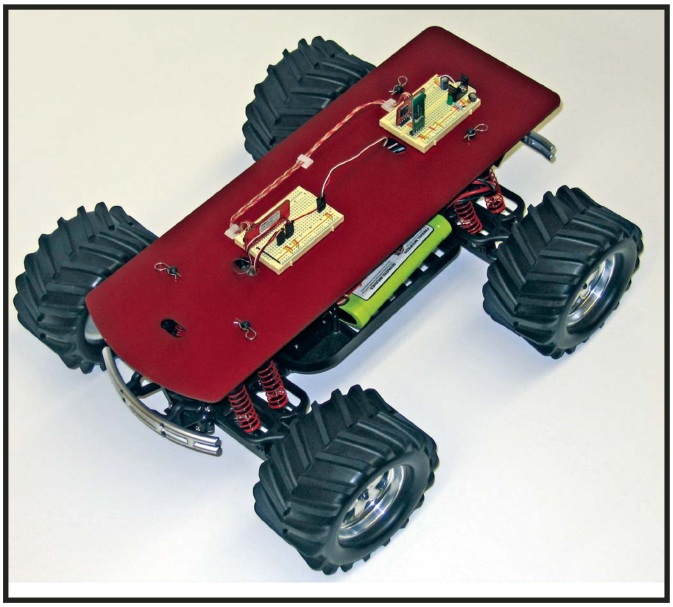
steer straight ahead without veering off-course.

When receiving input from a joystick, there are a couple of things you may have to handle in your code. If your controller is producing erratic output at the center position, you will need to ignore a small part of the range of motion near center. This is called the dead band. The way to handle it is to treat any axis values within the dead band as if the joystick was perfectly at center. A typical value for the dead band would be 10-20 percent of the full range.

Filtering can compensate for controllers that produce erratic output right through the full range of motion. Filtering averages the output values over a small period of time, compensating for such deviations.

The Logitech Extreme 3D Pro joystick did not require filtering, and the dead band could be set to 10%. Testing was also done using a Logitech Rumblepad 2. It had erratic output at center and throughout the range of motion, most likely due to wear. Using the Rumblepad 2 would require me to implement a dead band and filtering. All of the logic to process joystick events and translate them into servo positions is accomplished in the `convertJoystickToServoPosition()` method shown in the sidebar.

Figure 6. What the E-Maxx should look like at the completion of this article.



Taking Control

If you've followed the instructions from this and April's article, you can now steer, shift gears, and drive forwards and backwards. And your control system is robust and ready for even more additions and modifications.

The SDL project, referenced in the "Receiving Input from a Joystick" section, provides a great starting point for working with joysticks. If you want to use a Rumblepad 2, you'll have to enhance your code to implement filtering, as well. And you may also decide to explore some of the other SDL features to create a cross platform heads-up display.

While the ESC is great for high speed, off-road adventures, it's not particularly well suited for robotics. Trying to move at slow speeds is jerky at best and would make it difficult to get solid sensor readings. In my next article, we'll replace the ESC with motor controllers that will give me more fine-grained control over speed. We'll also add an encoder and start getting some feedback sent to the host module from the E-Maxx. **SV**

AUTHOR BIO

Chris Cooper is currently a software architect for Chicago-based Machine Bus Corporation. He has a B.S. in Computer Science from the University of Illinois, has presented at the OMG's Robotics SIG on Distributed Control Systems, and is a member of the Chicago Area Robotics Group (Chibots). He can be reached at cooper@coopertechnical.com

Demonstrations of Multipath Interferences

The Huygens Probe on Titan and a Simple Laboratory Experiment

by Ralph D. Lorenz

Scientists and engineers were amazed when the Huygens probe landed on Saturn's moon Titan and continued to transmit data for over an hour. With no idea what Titan's surface was like, no one had dared hope that it might survive impact for more than a few minutes. But after the excitement of looking at the images was over, careful study of the data revealed some unexpected surprises ...

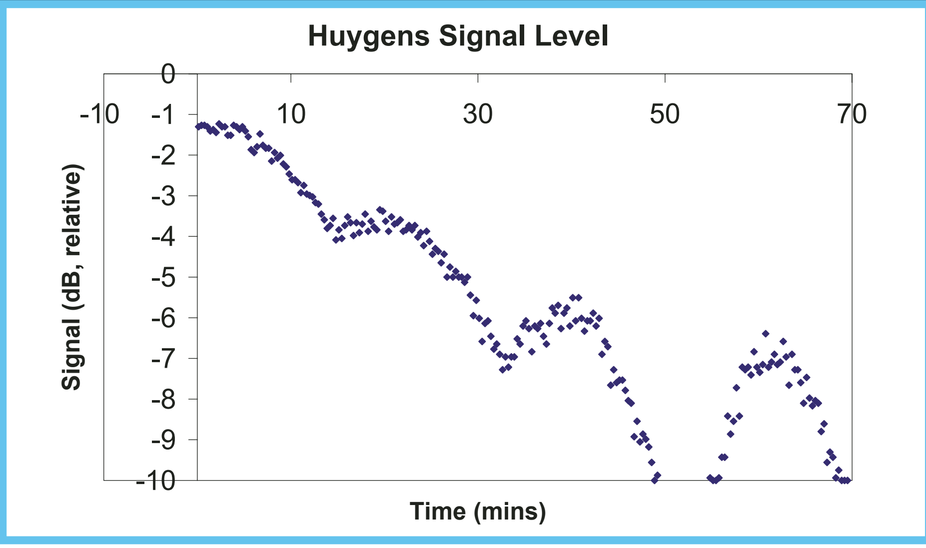


FIGURE 1. The strange behavior of the Huygens radio signal. The Cassini orbiter receiving the signal was slowly setting in the sky, its elevation falling from 20 degrees to zero over about 70 minutes, while the received signal strength went up and down several times (superimposed on a steady decline due to the changing distance between Cassini and Titan, and due to the poorer antenna gain at low angles).

The Cassini spacecraft that relayed Huygens data back to Earth also measured the strength of the signal. As expected, it fell as Cassini got lower and lower in the sky as seen from Huygens. However, as Figure 1 shows, it didn't fall smoothly, but went up and down in a bizarre way.

It was soon realized that what was happening was that at these low angles, with the radio beam just above horizontal, the echo of the radio signal (transmitted in all directions) from the ground was interfering with the direct ray path. In essence, the Cassini orbiter was flying through the interference pattern made by the constructive and destructive addition of the two signals (Figure 2).

This interference behavior is called 'multipath,' and is a well-known and troublesome problem for communications engineers. If you have used a TV with a set-top antenna, you may sometimes notice the signal degrading when somebody moves around the room. This is usually because they are reflecting the signal to create this interference. Cell phone reception can be similarly sensitive to position.

Recently, Miguel Pérez-Ayúcar and I, working with other engineers from the European Space Agency analyzed[1] the signal strength pattern from Huygens and were able to deduce some of the scattering properties of the Titan surface. In effect, we got a crude radar experiment for free!

Even though this effect was seen in a multibillion-dollar space mission, it is possible to recreate the effect with about $10 of electronic parts, using ultrasound instead of radio. Ultrasound at a few tens of kHz has a wavelength of around 1 cm, so the interference geometry seen by Huygens (with its radio wavelength of ~15 cm, and its antenna about 75 cm off the ground) can be seen by mounting the ultrasound 'antenna' a few cm above a reflecting surface.

The ultrasonic transducers are common and inexpensive items (about $7 — I used part # 139491 from Jameco; **www.jameco.com**), used widely in mobile robot applications and as tape measures. They are usually obtained as a transmitter/receiver pair. Although they look similar externally, it is important to use them the right way around!

They are tuned to operate best at 40 kHz. A signal generator or a microcontroller can generate a suitable driving signal. Alternatively, a very simple oscillator using a 555 timer also performs adequately. A trimmer potentiometer in the circuit is used to adjust the oscillator frequency to match the resonant frequency of the transducer

Reference

[1] Pérez-Ayúcar M., Lorenz R. D., Floury N., Prieto R., Lebreton J-.P., Surface Properties of Titan from Post-Landing Reflections of the Huygens Radio Signal, in preparation.

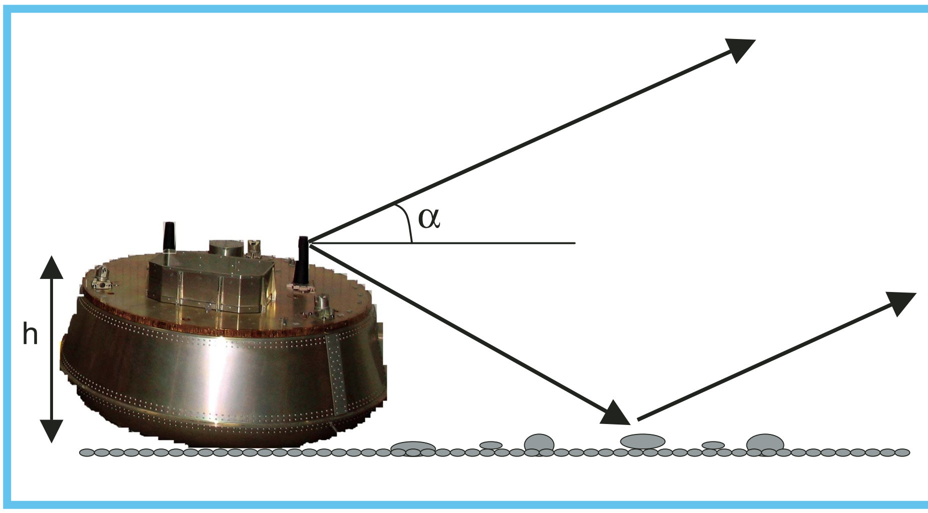


FIGURE 2. When seen from a large distance away at an elevation angle α, the direct and reflected beams are not seen individually, but added together. There is a phase difference between the two beams, of $\delta = h*(1-\cos(2\alpha))/\sin(\alpha)$. When the phase difference — taking any phase reversal on reflection into account — is equal to an integral number of wavelengths, the direct and reflected beams add together in phase, producing a stronger signal. Halfway between the angles at which this occurs, the two signals are out of phase and at least partially cancel out. The extent to which this happens depends on how reflective the surface is.

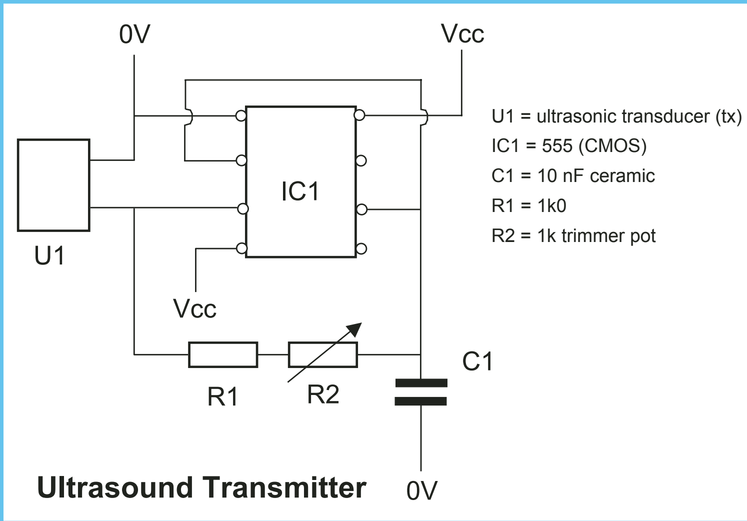


FIGURE 3. A 40 kHz oscillator circuit. Although a pure sine wave would be ideal, the square-wave oscillator above seems to work perfectly well.

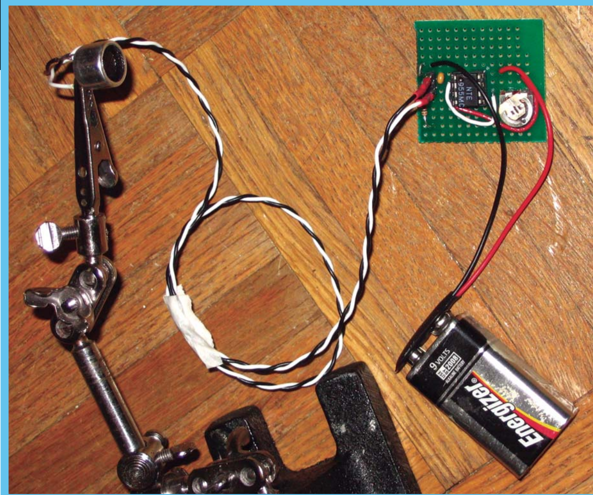


FIGURE 4. The transmitter transducer held a fixed distance above the floor with a small clamp. The battery-driven oscillator circuit is on the small circuit board.

to give the strongest signal. Figure 3 shows the schematic for this circuit.

At short ranges, the receiver output is enough that it can be monitored directly using an oscilloscope. However, longer range and more versatility can be had by amplifying the output. A range of op-amp circuits can be used, but to minimize the parts count I used an LM386 amplifier. For conveniently monitoring the signal strength with a voltmeter, the amplified output is rectified with a diode (see Figure 5).

With no signal (the transmitter switched off), the voltmeter on the receiver read about 3.8 V. Just by holding the transducers away from the workbench and aiming them at each other, it can be seen that they are moderately directional, with their response varying smoothly over a 30 degree half-beamwidth or so.

With the transmitter mounted 8 cm or so off the ground, the receiver was set up 150 cm away, using a laboratory stand and clamp to adjust its vertical position. The transducer was aimed horizontally, with no attempt to point the transducer up or down.

Moving the receiver up and down gives a very different response from the free-air gain pattern. There are very sharp nulls: a movement of less than a centimeter can give a dramatic change

FIGURE 5. Receiver circuit. If you use an oscilloscope to measure the output, D1 should be unnecessary.

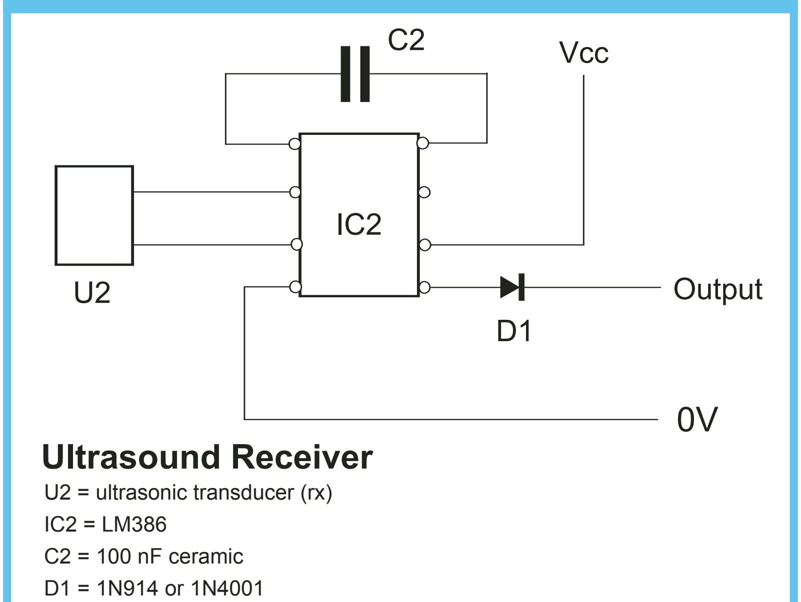
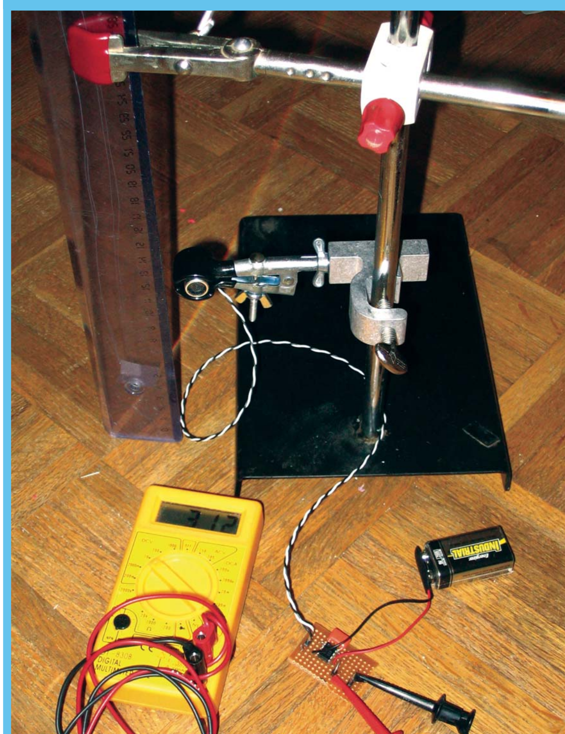


FIGURE 6. Ultrasound receiver. The transducer was held in a clamp at various heights above the floor, and the signal amplitude monitored with a digital voltmeter and the receiver circuit below.



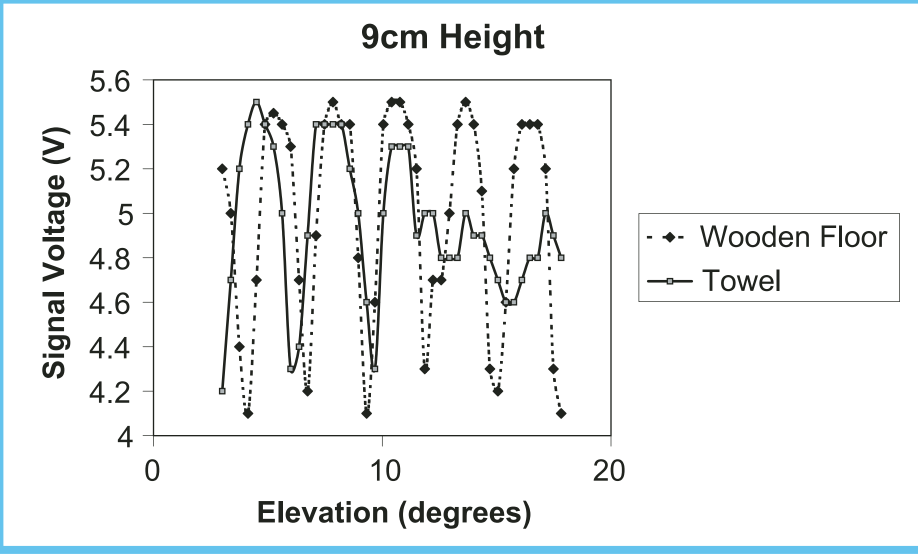


FIGURE 7. A strong and constant interference pattern is seen with a strong reflector like a wooden floor. The pattern breaks down for a rougher and more absorbing surface like a towel.

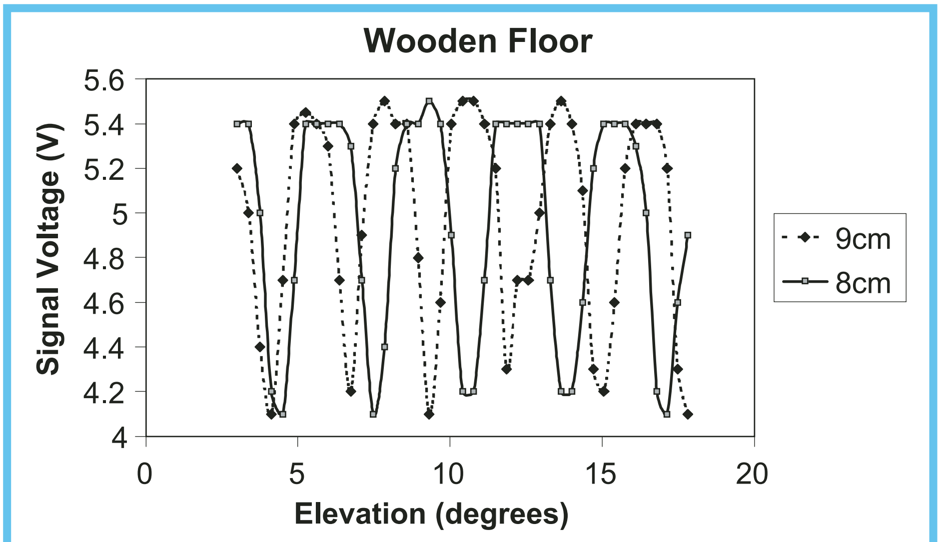


FIGURE 8. An interference pattern shifts in angle for even a very small change in height of the transmitter. The amplitude of the pattern stays about the same, however.

in the received signal strength. The dips in the beam pattern seem to be narrower than the peaks in the received signal strength.

The nulls are very sharp when the surface between the transmitter and receiver is a good mirror for sound (a so-called specular reflection, from a dense, flat surface). However, if a more absorbing surface is placed there, the reflected signal is much weaker. That means it interferes less with the direct signal, and so the nulls and peaks are less prominent. You could try different surfaces — carpet, gravel, sand (perhaps sculpted into different shapes).

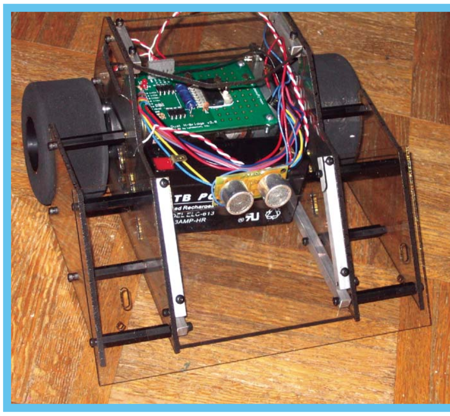


FIGURE 9. A typical small robot with an ultrasonic ranger mounted a few cm above the floor — as this article shows, such a location is prone to multipath effects.

The profile of signal strength with angle will indicate (albeit indirectly) the variation of echo strength with angle.

For example, the plot in Figure 7 shows a smooth wooden floor (a good reflector at all angles) has a very strong and constant interference pattern — the echo is strong enough to interfere completely with the direct signal (remember there is an offset voltage, so ~4 V means no signal). On the other hand, a towel spread across the floor shows a strong pattern at shallow angles, but a much weaker modulation (indicating a weak echo) at steeper angles.

Another effect can be seen on varying the height of the transmitter. As Figure 8 shows, this tends to shift the pattern of nulls. The effect is very sensitive: a significant shift in the pattern can be seen with only a small change in position. A fraction of a wavelength change in position causes a big phase difference between the direct and reflected ray and, thus, a shift in the interference pattern.

These results show that the performance of sonars mounted near the ground, as is typical on mobile robots, can be highly sensitive to multipath effects. Occasional poor performance may be encountered if the reflecting target happens to sit in a null formed by the interference of the direct beam with a reflected signal.

Some additional experiments can be suggested. One might be able to measure the speed of a moving reflector like a golf ball through the interference pattern by setting up a transmitter and receiver near a reflecting floor. As the ball flies through the beams, the reflection will increase and decrease in strength. Or a sonar could be mounted on a servo or similar pointing platform to change its position next to a reflecting surface, so that the beam pattern can be changed. The change in beam strength for a given angle will be much larger than would be expected by changing the orientation of a transducer or transducer pair in isolation. SV

ROBOTS: FROM SCIENCE FICTION TO TECHNOLOGICAL REVOLUTION

Book Review

by Tom Carroll

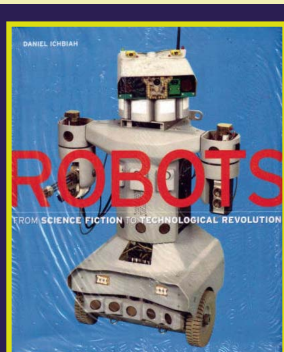
We all have enjoyed filling our private robotics libraries with those indispensable books that fit our particular niche in robot experimentation, such as Gordon McComb's *Robot Builder's Sourcebook* and his and other's many useful titles. Sometimes it is just fun to go out and get a book that's relaxing to sit down with for a good read. My wife, Sue, gave me such a book this year for my birthday. Entitled *Robots, From Science Fiction to Technological Revolution*, this book written by Daniel Ichbiah and translated from the French was published by Harry N. Abrams in 2005. It is a great book to curl up with on a rainy day.

There are many color photographs throughout its 540 pages that illustrate many of those robotic subjects that we all hear about on the Internet and in the news, but never get to know more about. The book is divided into 12 sections covering: The History of Robots, Robots in Fiction, The Androids, Domestic Robots, Robots in Industry, Robots as Explorers, Security Robots, Robots and Medicine, Playful Robots, Robots in the Arts, Robots in the Future, and a Brief Guide for the Robotics Enthusiast. It is written, neither as a text nor a reference book about any one aspect of robotics, but rather as an overall review of the subject of robots and robot history. Since the author is French, it is a bit slanted towards French and European robotics, but US and Japanese robotics are well covered.

The illustrations and photographs cover their intended subjects well and I found only a few errors in the references. Some of the translations seem to use words that we, in the US, would not use, but I find these interesting. The author also provides interviews with robotic experimenters in sideboxes throughout the book. Readers from many different levels of background in robotics will all find this book a very enjoyable read. Its list price is $37.50 but is can be purchased for as little as $24.75. SV

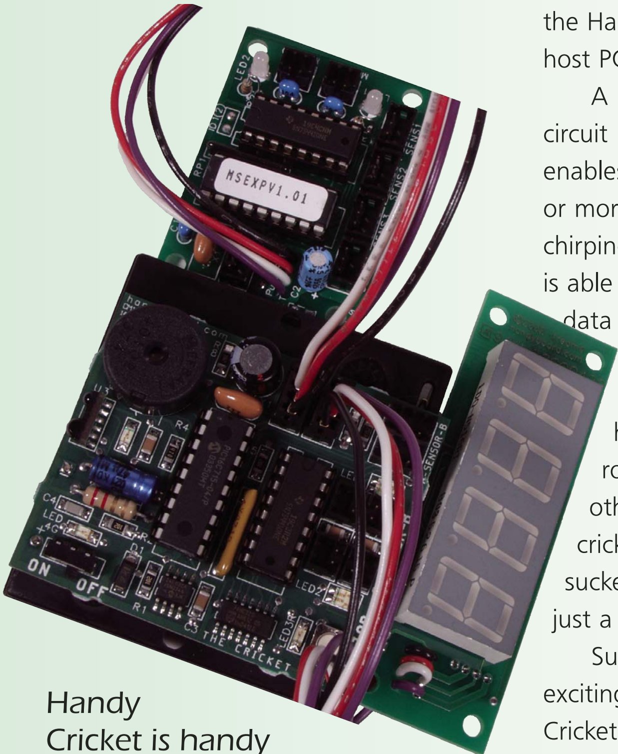
ROBOTS
From Science Fiction to Technological Revolution

- Hardcover: 540 pages
- Publisher: Harry N. Abrams, Inc., New York
- Published: September 20, 2005
- Language: English (Translated from the French)
- ISBN #: 0810959127



Your Next Robot Can Be a Lot Chirper

Is there any robot builder who doesn't know about the 68HC11-based micro-controller *Handy Board* designed by MIT?



Handy Cricket is handy indeed — for controlling compact robot designs. A variety of optional interfaces can be plugged into the Handy Cricket.

In a remarkable gesture to the robotics community, MIT has licensed the Handy Board design as "freeware" for educational, research, and industrial use. While you can readily download design information about building your own Handy Board on the web (**handyboard.com**), there are a couple of manufacturers who sell assembled Handy Boards.

Since 1995, Gleason Research (**gleasonresearch.com**) has been selling the MIT Handy Board to robot builders worldwide. Now, a smaller "freeware" design is available and it is perfect for installation inside compact robot designs.

The Handy Cricket Version 1.1 is a low-cost module based on the Microchip PIC® microprocessor featuring a built-in Logo interpreter. Equipped with two motor ports, two sensor ports, two bus ports, 4 Kb of static memory, and a piezo speaker, the Handy Cricket connects with the host PC via a serial port IR interface.

A unique IR transmitter/receiver circuit built into the Handy Cricket enables communication between two or more Handy Crickets. Get it, like the chirping of a cricket, the Handy Cricket is able to "chirp" IR signals at a 50K data rate between another Handy Cricket.

Imagine this, you could have various Handy Cricket robots "talking" between each other. And just like a biological cricket, the Handy Cricket is a tiny sucker. The overall dimensions are just a bit under 2-1/2 inches per side.

Sure all of this hardware stuff is exciting, but the part of the Handy Cricket that leaves me salivating is the fantastic implementation of the Logo programming language, called "Cricket Logo," that is built into the Handy Cricket.

The Handy Cricket is programmed in a language that is a simplified version of the powerful, yet easy-to-learn Logo language. Unlike most programming languages, Cricket Logo (as well as its ancestral Logo) is short, simple, sweet, and primitive. I'll grant you, Logo is a little archaic, but the payback can be terrific for robot builders. For example, commonsense commands like BRAKE and SETPOWER don't require a lot of human smarts to figure 'em out.

Individual Handy Crickets cost $59 or, you can purchase a complete Handy Cricket Starter Kit for $99.

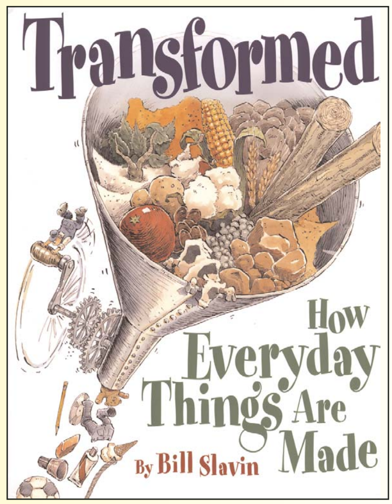
A Touchy Feely

Are you looking for a great touch sensor?

The QTouch™ family of touch and proximity sensors from Quantum Research Group have the ability to sense physical contact through plastic or glass up to 100 mm (4") thick. Even gloves won't stop QT110, QT111, QT112, QT113, QT115, and QT118H from detecting a touch. Even better, QT113 and QT118H are also able to sense moisture. So sweaty robots, beware.

The QT11x family of chips

So That's How It's Made ...



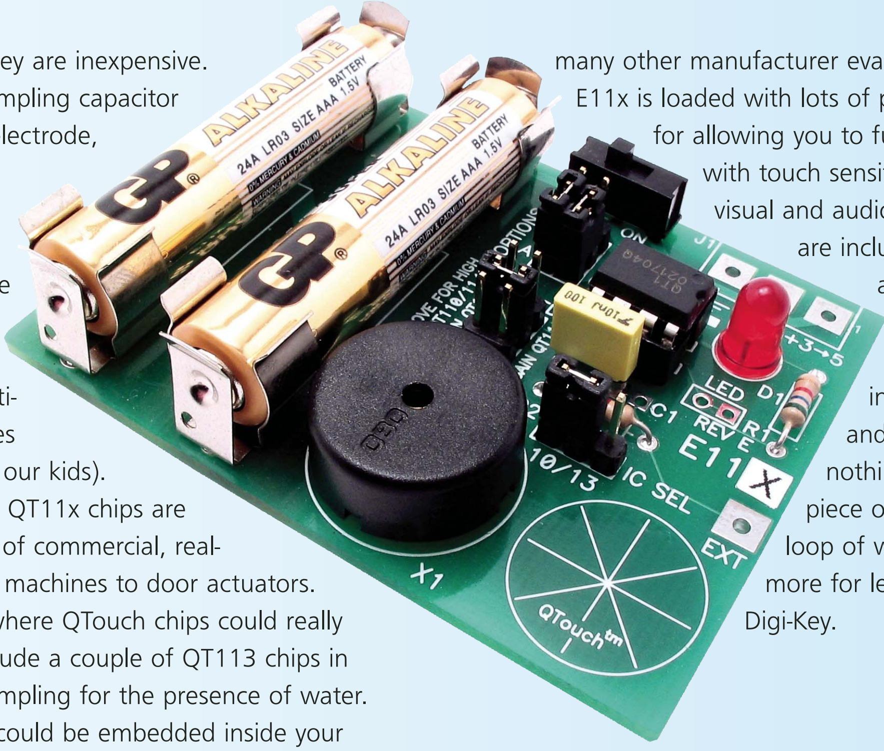
If you've ever wondered how they make plastic dinosaurs, you're not alone.

Bill Slavin has written a book that actually illustrates the complete manufacturing process for this novelty, as well as 68 other common household products. *Transformed: How Everyday Things Are Made* (Kids Can Press, Ltd., 2005; 160 pgs; $24.95) can give you some great insight into how you should be designing and building your robots.

are as easy to use as they are inexpensive. Just slap an external sampling capacitor on it, add a detecting electrode, and zap it with around +3V and you have the perfect robot touch sensor. Furthermore, the chips all feature auto-calibration, drift compensation, and automatically calibrate themselves after a time out (unlike our kids).

As for applications, QT11x chips are found in a wide variety of commercial, real-world devices from pay machines to door actuators. In robots, however, is where QTouch chips could really shine. For example, include a couple of QT113 chips in your biped's feet for sampling for the presence of water. Or, an actuator switch could be embedded inside your chassis for receiving outside input without having to drill a hole for external "feelers" in its body shell.

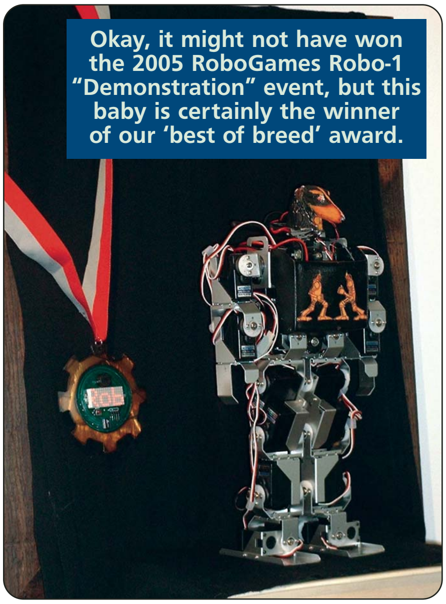
If you're looking for a painless way to experiment with the QT11x family of touch sensors, then look no further than Quantum's E11x QTouch Evaluation Board. Like so many other manufacturer evaluation boards, the E11x is loaded with lots of powerful features for allowing you to fully experiment with touch sensitivity sensors. Both visual and audio touch indicators are included with the E11x and you can easily plug your own external electrode into the board. Oh, and an electrode is nothing more than a piece of metal foil or a loop of wire. All of this and more for less than $20 from Digi-Key.



Unlike some evaluation boards, the Quantum E11x Evaluation Board can be dropped straight into your robot design adding some extra touch sensing capabilities for under $20. (Photograph courtesy of Quantum Research Group)

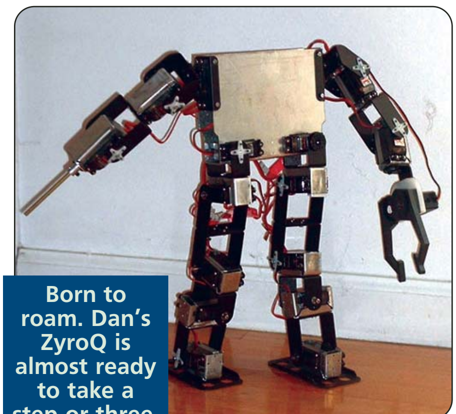
MENAGERIE

Builder's Name: Daniel Albert
Significant Robot Building Milestone: Remote WiFi controlled KHR-1 via PDA
Robot's Name: RoBorzoid
Robot's DoB: 01/30/2005
Robot's Height: 15"
Robot's Weight: 3 lbs
Significant Robot Living Milestone: third place at 2005 RoboGames Robo-1 "Demonstration" event
Additional info: Yes, the 'B' in RoBorzoid is capitalized. It's part of his name which is a melding of Robot and Borzoi. Borzoi is the breed of dog the head is cast from.

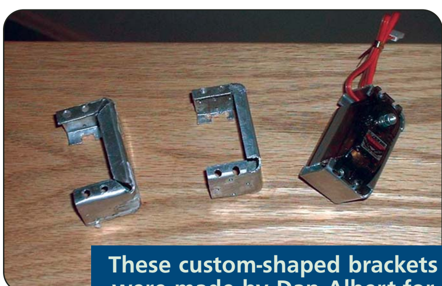


Okay, it might not have won the 2005 RoboGames Robo-1 "Demonstration" event, but this baby is certainly the winner of our 'best of breed' award.

According to Dan, "I will soon make my custom version of the 'HeartToHeart' control program available for free download. This is the one I used at RoboGames. It runs on any PDA that runs a version of Windows CE. It has most of the 'HeartToHeart' features and yet is portable and uses both Bluetooth and WiFi 802.11 protocols."



Born to roam. Dan's ZyroQ is almost ready to take a step or three.



These custom-shaped brackets were made by Dan Albert for his ZyroQ robot.

www.alberts-equation.net
www.sjrobotics.org

Builder's Name: Daniel Albert
Significant Robot Building Milestone: Home built using some brackets from Lynxmotion and 16 homemade servo brackets. Remote control via Bluetooth and a custom control program written in C++.
Robot's Name: ZyroQ
Robot's DoB: 10/18/2005
Robot's Height: 15"
Robot's Weight: 4.5 lbs (he's currently on a diet to shed some weight)
Significant Robot Living Milestone: It can almost walk!

Robot Builder Bio:

Dan earns a living as an embedded programmer. His 30+ years of experience in programming computers aids with his robot designing and building.

Dan is also president of the South Jersey Robotics Group that hosts the "RoboPraxis" annual robotics event.

Want to Be Famous?

Okay, here's your chance to get your name and your robot's name up in lights; well, at least published inside *SERVO Magazine*.

First of all, here is just a little bit of photography advice. Please take 300 dpi imagery that is clear of distracting backgrounds without any weird distortion(s). Ensure that you use good, even soft lighting; direct flashbulb glare can be very distracting and wash out valuable robot details. Also, take as many photos as possible — showing details, construction steps, final views, etc. Remember, a great photo is worth about a thousand words; or something like that.

However, feel free to send words also, especially on the purpose of your bot, technical specs like processors used, unique design elements, and whatever else you think might be interesting. Bundle your images and captions along with this valuable information:

Builder's Name:
Significant Building Milestone:
Robot's Name:
Robot's DoB:
Robot's Height:
Robot's Weight:
Significant Robot Living Milestone:
Website URL (if you have one)

You can email your submission to: menagerie@servomagazine.com. Or, if you have too many photographs, you can mail us a CD-R to: Menagerie, SERVO Magazine, 430 Princeland Court, Corona, CA 92879-1300.

COMBAT ZONE

Featured This Month

Participation

44 *Robot Combat Event Safety Primer* **by Kevin Berry**

45 *So You Built a Bot, What's Next?* **by Kevin Berry**

Feature

46 *Designing and Building a 12 lb Fighting Robot* **by Peter Smith**

Events

47 *Upcoming*

49 *Results*

Technical Knowledge

48 *Designing a Combat Robot Drive Train* **by Steve Judd**

Product Review

49 *Scorpion ECSs* **by Kevin Berry**

PARTICIPATION

Robot Combat Event Safety Primer

● by Kevin Berry

Rule #1 — Don't do anything stupid or dangerous.

Rule #2 — In the absence of checks and balances, even the most veteran builder will violate Rule #1.

I have been the "safety puke" at many small combat events. While most folks think that "newbies" are at the greatest risk — and there are a few careless ones — in my experience, it's the veterans, eager to get back in the box after a frantic damage repair session, that tend to violate basic safety rules.

There are two main areas of safety at events: Pit Safety and Robot Safety. Pit Safety, involving mostly tool use and industrial type accidents, might be the topic of another column. Here, I'm going to introduce the basic elements of Robot Safety, as detailed in the Robot Fighting League's standard rule set. Like most rule sets, there's a scar for every rule!

All bots should have a power switch or link, and a positive indicator light when bot power is on. A separate switch/light for drive and weapon power is even safer. All bots, whether five ounces or 340 pounds, should have their wheels off the table or ground at all times except when in the box just before a fight. Weapons must have covers on sharp edges, and positive restraints, except when in the box. Restraints come off last thing before a fight, and go on first thing after. Radios MUST ONLY BE POWERED ON after checking out a unique frequency clip from the controller. Too many times I've seen someone accidentally turn on a radio and energize someone else's bot.

Lastly, THINK and WATCH! There are myriad examples of builders preventing others from getting hurt, doing something stupid, or observing a problem with an unattended bot and taking positive action. Rule #2 mentions "checks and balances." This puts the onus right on the builders, not just the overworked official safety puke! **SV**

So You Built a Bot, What's Next?

● by Kevin Berry

Attending your first combat event can be a bit overwhelming. This article is meant to help you ease into the sport without facing any more stress than a veteran. In fact, "newbies" often are better prepared than the old hands, some of whom are building in their cars on the way there!

Getting Started

The best way to find out about an event in your area is by watching UI Production's Builders Database, **www.buildersdb.com/events.asp** Since virtually all North American combat events use this site, this article is based on their system. When you see an event you'd like to participate in, email the Event Organizer (EO) and introduce your team. Don't be shy, EOs are generally ecstatic to have new builders. Let them know who's on your team, describe your bot, make sure they know you've read the tech regs and event rules, and that you'd appreciate any tips or help they can give. Many EOs will try, if asked, to hook you up with a veteran builder so you have a mentor at the event.

Signing Up

The first step is to build your team and register your bot(s) on the database. Then, you can sign up for your event. Be aware that most EOs set a maximum number of team members per bot. If you need to bring more people, (e.g. a family of four with one antweight bot; a common starter team), contact the EO. Large events that charge an admission fee may ask you to buy a ticket for some of your members. You will (usually) be charged a registration fee per bot, and asked to download, sign, and mail a participation agreement. There is no better way to start on the right foot with an EO than paying and mailing paperwork promptly!

Getting There

When packing for the event, you'll need to remember a couple of things. First, bring everything you need to rebuild your bot, almost from scratch. Conversely, remember you'll probably park a LONG way from the pits, and have limited table space. Most teams bring a dolly or hand truck. You may be asked to bring your own table, and there are rarely enough chairs, so bring your own. Sometimes there is an unloading zone close to the pits, with parking elsewhere. This is a typical question to ask ahead of time, so include it in your initial email if it's not on the website.

Checking In

If you get to an event early, they may not be ready for you quite yet. If that happens, immediately ask: "What can I do to help?" If you arrive on the late side, you may find mass confusion. Buttonhole the first person you see, and ask for an introduction to the EO or registration table. Someone will point you to a pit table, where you can set up. Once you're unloaded, you need to find the weigh-in station. There your bot will be weighed. Come to the event KNOWING your bot is under the weight limit. Many people spend the first hour at an event hacking pieces off their bots.

The next thing you need to find is the Frequency Board. There you'll see clips (usually clothespins) labeled with the allowable R/C frequencies. Make sure your bot's frequency has a clip. You may be given an "exchange clip" with your bot or team name, to leave when you take a clip. The absolute top rule of safety at any event is to never turn on a transmitter without a clip. You could hurt someone seriously if you power up another bot by accident!

The next step is to "safety" your bot. This procedure, if not covered in the posted information, is another thing to ask about ahead of time. Usually it involves checking out a frequency clip, putting your bot in the arena (under the supervision of the event's Safety Officer), making sure the weapon and drive system stop when the radio is turned off, and an inspection of your weapon restraints or covers. Again, you should *know* you will pass safety before you come to the event.

Once you've passed weigh-in and safety, you'll be entered onto the fight brackets. Now is a perfect time to make the rounds and introduce yourself to other builders. They'll be busy, but glad to see you. One of the backwards things about this sport — it's considered polite to talk strategy with your opponents, discussing ways to beat each other ahead of time!

Pit and Fight Etiquette

"Always A Borrower And A Lender Be." Cooperation in the pits is the hallmark of this sport. Don't be afraid to ask for help or tools, but always return borrowed items promptly, and be very forthcoming loaning your tools and spare parts. Sportsmanship in the box is highly respected, which means granting postponements graciously, not taking cheap shots on an obviously disabled opponent, and even (sometimes) freeing a stuck bot so the fight can continue. (The last is a function of your confidence in beating them, and the tone of the event). Shake hands with your opponent after the fight, and thank the referee.

Robot Combat is a sport that encourages cooperation and sportsmanship, while prizing aggressive fighting. Starting off humble, friendly, and competitive is the best introduction to your peers, and will result in quick acceptance and lots of help. SV

DESIGNING AND BUILDING A 12 POUND FIGHTING ROBOT

● by Peter Smith

The 12 lb "Hobby Weight" fighting robots are one of the most popular and fastest growing classes in this exciting sport. They are big enough to look and even sound like the bigger bots that appeared on the TV shows while being much cheaper to build and enter into events.

The first stage in designing any bot is to look closely at the competition. Go to events, if possible, to see what works and what doesn't. Study the rules for the class; most events use the Robot Fighting League (RFL) rule set (available at **www.botleague.com**) and probably most important of all is to visit the RFL's forum, starting at **www.botleague.com/forum.asp** to get first-hand advice from real builders.

There are many other websites that offer specialist robot parts, **www.banebots.com**, **www.robotcombat.com**, and **www.battlepack.com**, for example, and others like **www.cncbotparts.com** and **www.teamwhyachi.com** that can provide specialty parts and also specialist machining and other engineering services. Finally, there are even websites like **www.inertialabs.com** and **www.rollingthunder.com** which offer battle proven designs in kit form.

The next step is to decide what type of bot to build. It is tempting to jump straight in and try to build a bot with a big spinning blade or a pneumatic flipper, but I would advise any first-time builder to start with either a "Wedge" or a "Brick." There are several reasons for this. Firstly, they are fundamentally simpler that any robot with an active weapon, therefore easier to design, build, and maintain. Secondly, they are safer and much less likely to cause injury to a young or inexperienced builder or driver. Lastly, they tend to be tougher and are more likely to keep working during fights and survive to fight another day.

To compensate for the lack of a weapon, you will need to combine good speed, power, maneuverability, and armor. This requires careful choices of drive motors and gearboxes, electronic speed controllers (ESCs), batteries, radio control equipment, and finally what materials to use for the chassis and armor.

Drive Motors and Gearboxes

These combine a small electric motor with a gearbox. Good choices would be converted cordless drill motors (good cordless drills are available for as little as $16 from **www.harborfreight.com** or already converted from **www.teamrollingthunder.com**), or the 25:1 36 mm 545 motors from BaneBots or the TWR18 from Team Whyachi. All of these will give good speed and power when combined with 3" to 4" wheels and the right battery pack.

Electronic Speed Controllers

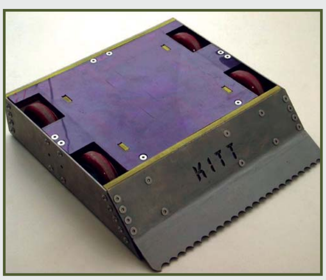
The BB-12-45 from BaneBots or the Scorpion XL from **www.robotpower.com** are popular choices. The latter is an all-in-one unit that combines two ESCs and a mixer on one board. You would need to add a mixer (or mix using a computer transmitter) when using a pair of the BB-12-45s, but their small size lets them fit in tight spaces.

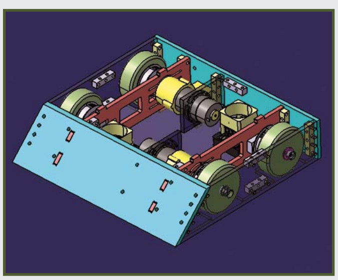
Batteries

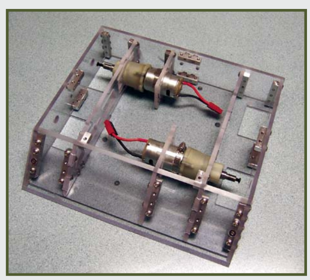
Packs of 14.4V–18V (depending on your choice of motor voltage) GP2000 NiMH cells are ideal for this weight class, giving plenty of current when required and enough capacity to last the longest fights. These are available from **www.battlepack.com**. It is possible to use cheaper hobby packs connected in series, such as those sold for R/C cars, but they tend to be heavier for the same performance. You will also need a suitable charger for your pack.

Radio Control

While many events will allow you to use cheap 27 MHz AM and FM transmitters on non-weapon hobby weights, it is probably better to get a 75 MHz FM set (you will not be allowed to use 72 MHz R/C aircraft equipment). Added features are built-in computers that allow mixing and







also PCM, which is a requirement for most bots with weapons.

Materials

0.25" polycarbonate (one brand name is "Lexan"), 0.25" UHMW, and 3/16" aluminum are popular choices for chassis materials and all are easily cut, drilled, and tapped using hand tools. 0.08" – 0.1" thick titanium is the ideal choice for armor. It too can be hand cut and drilled, but this takes more skill and better equipment.

It is vital that you keep an accurate track of the weight of the parts you choose and design. Many excellent designs have had to make serious compromises when they discovered that they were over the weight limits just before their first competition. A computer spreadsheet program like Excel is ideal for doing this. The actual part designs can be done on paper or on one of the many computer-aided design (CAD) programs that are available. I usually use a professional grade package called SolidWorks which I can recommend, but there are many other that are much cheaper and that will do the job almost as well.

Do your research first, don't be afraid to ask questions, and listen to the advice offered. There is a good chance that your first bot will be successful. One last piece of advice: Try and complete your build well before your first competition and get lots of driving practice. Good driving has won more fights than great design! **SV**

EVENTS

Upcoming (all RFL National qualifiers)

Central Illinois Robot Club — Central Illinois Bot Brawl, Lakeview Museum, Peoria, IL 5/6/2006 **http://circ.mtco.com** 1 lb R/C Combat, Autonomous Sumo (3 kg, 1 kg, 500 g, LEGO), Spectators free. Teams include high school students, adults, and a mix of genders. CIRC has been holding Sumo competitions since 2002, and combat since 2003.

Ohio Robot Club — House of Robotic Destruction Spring 2006, Olmsted Falls Community Center Olmsted Falls, OH 5/20/2006 **http://www.ohiorobotclub.org/** This event is for Fairy, Ant, and Beetle weight combat robots. This is ORC's third event, and attendance is growing each time.

Mechwars Robot Combat — MECHWAR-9 will be held May 20-21 at the Eagan Civic arena, 25 minutes from downtown Minneapolis. **www.tcmechwars.com/**. Spectator tickets are $6 per day. Mechwar-9 is the 13th large-scale event put on by the Mechwars crew since 1999. These tournaments are unique in that judging is based on a series of highly quantifiable "damage" events with assigned point values. Fire is encouraged, and stalled or turtled bots are restarted after taking 20 seconds of punishment. The arena measures 40 x 32 feet, with hazards including an 80 pound steel bar rotating at 4,500 RPM, and a 2,000 degree flame pit. The Mega-weight class (390 lbs) constitutes the very largest fighting robots currently battling anywhere in the world. Mechwars is also unusual in its emphasis on spectator turnout and showmanship. With a spectator high of 2,620 tickets sold for a single event (30,000 over 10 days of State Fair events), Mechwars specializes in getting the teams in front of plenty of fans. Typical teams are made up of engineers, science fiction fanatics, shade-tree mechanics, college students, Girl Scout troops and the occasional crazed loner, a Minnesota specialty. Mechwar-9 will be a qualifier for the RFL National. The unique prize structure ensures that every functional entry will win some prize money, with weighted purses for top finishers. Trophies are awarded to first through third place. All Mechwars tournaments are filmed and event DVDs are available to both teams and the public.

California Insect Bots — Gilroy Bot Gauntlet, Hobby World, Gilroy, CA 5/27/2006 **http://calbugs.com/** This is open to Flea weights, Ant weights, and Beetle weights. Organizer Dave Wiley's description of the sport: "Gratuitous medieval violence with none of the guilt." This event has been held annually since 2004, and teams are a mixture of high school and college students, families, and adult professionals.

Whyachi House of Robotic Entertainment — Spring W.H.R.E '06, Dorchester, WI 5/27/2006 **http://www.whre.org/**. No pit passes required, no limits on number of pit members, no fee for spectators. Held at Team Whyachi's bot shop, hosted by veterans Terry Ewert and Dick Stuplich. 150 g through 220 pound classes.

TECHNICAL KNOWLEDGE

Designing a Combat Robot Drive Train

● by Steve Judd

When designing an electric drive train for a combat robot, you will quickly find yourself asking three questions:

- How fast will it go?
- How much "pushing power" will it have?
- How much battery power will it need?

With your design in hand and some information about the available motors and batteries, you can answer all of these questions — but you'll need to do a little math.

For every permanent magnet DC (PMDC) motor, there is a fixed relationship between RPM and torque. This is expressed in two constants, called "motor constants." These are the voltage constant (Kv) expressed as RPM/volt and the torque constant (Kt), usually expressed in in-oz/amp. For all PMDC motors Kt * Kv = 1355: if you know one constant, you can derive the other. The maximum torque produced by a PMDC motor occurs at 0 RPM: the motor is *stalled*, and the current it draws is the *stall current.* If you multiply the stall current by Kt, you will find the maximum torque.

If you know the motor constants and stall current for your motors and you know how much torque you need, you can determine your battery requirements.

First let's explore some of the design decisions you'll need to make.

How Fast is "Too Fast?"

A common misconception is the idea that faster is better. The largest arenas are only 30-40 feet across, and smaller arenas of 16 feet or so are more common. There is not much point in an "impressive" top speed if there isn't room in the arena to reach it.

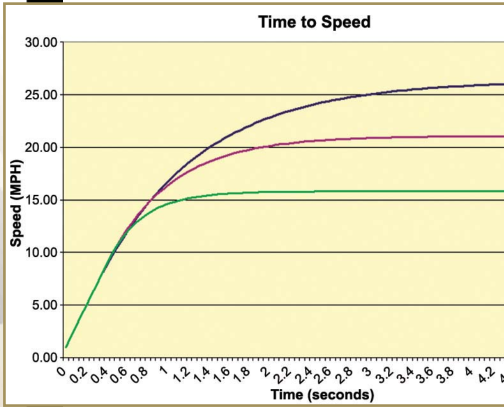


FIGURE 1

FIGURE 2

Figure 1 shows the time to reach top speed for a 30 lb robot powered by two 18V DeWalt drill gearmotors at 24V, with three different wheel sizes: 3", 4", and 5".

The 5" wheels deliver a blistering 26 MPH top speed, but it takes over four seconds to achieve it. The 4" wheels yield 21 MPH in 2.6 seconds, and 3" wheels 15.8 MPH in 1.7 seconds. Note that for the first .8 seconds, all three are almost exactly the same.

Figure 2 shows the *distance* covered by each wheel size on the same time scale. There isn't enough room in the arena for the increased speed to make a difference. For the first 1.2 seconds, all three cover the same distance — about 20 feet. The smaller wheels provide more torque and thus faster acceleration, at a cost in top speed — but we can't actually **use** the additional speed, so it doesn't matter.

Torque, Traction, and "Pushing Power"

The torque produced by your drive train accelerates your robot and provides the "pushing power" in a shoving match. You can never have "too much" torque, but as with speed, once you have "enough" having more doesn't pay many dividends.

Traction limits the usable torque of your robot. The maximum "pushing power" of a given robot is the robot's weight multiplied by the coefficient of friction (Cf) of the tires. For common robot tires, a value of **.9** is a reasonable value for Cf.

A 30 lb robot can apply a total of 27 lb-in of torque to the floor before the wheels break traction and spin; the same robot carrying its opponent can apply 54 lb-in of torque to the floor before breaking traction. More torque than this cannot be applied to the floor and represents an unusable resource — thus "enough" means *enough torque to break the wheels loose while carrying your opponent*.

Determining Power Consumption and Choosing Batteries

Our example robot is a 30 lber with two DeWalt 18V drill gearmotors and 3" wheels. In high gear, these motors have a Kv of 73.74 and

a calculated Kt of 18.37 oz-in/amp. The *stall current* for these motors is 155A, so the maximum torque available is 155*18.37: 2847 in-oz, 177 lb-in. The wheel diameter is 3”. This gives a reduction of 1.5:1 (1.5” being the wheel radius), so the maximum torque at the point the wheel touches the floor is 118 lb-in. This is much more than the torque needed to spin the wheels while carrying the opponent — 54 lb-in. You have more than enough torque, how much current will your motors draw in a match?

You know the “wheelspin torque” for the robot is 27 lb-in in normal driving (e.g., not carrying the opponent). The reduction at the floor with 3” wheels is 1.5:1 and there are two motors, so the wheelspin torque for each is 27*1.5/2: 20.25 lb-in, or 324 oz-in. The Kt is 18.38, so each motor will draw 17.63A (324/18.38) at wheelspin for a total of 35.25 amps. This is the *peak discharge* — the amount your batteries must be able to supply on demand.

If you were to drive at the very edge of wheelspin for a three-minute match, you would use (3/60)*37.25 or 1.86 amp-hours. This is your minimum *capacity* — the amount of power your batteries must contain.

Battery Characteristics: Capacity and Discharge Rate

Battery manufacturers often give the discharge current as a multiple of capacity (C); “15C” means the battery will provide a current equal to 15 times the rated capacity. A battery rated “1 AH, 15C” can be expected to produce 15A. The battery specifications may distinguish between “continuous” and “peak” current. “Peak” is what the battery can source for a very brief period, “continuous” is what it can deliver continuously.

Different battery chemistries have different discharge characteristics. Sealed Lead Acid (SLA) batteries can provide far less than the rated capacity (as little as 40%) under conditions of very high load. Nickel Cadmium, Nickel Metal-Hydride, and Lithium Polymer batteries typically provide lower discharge currents but up to 90% or more of their rated capacity under high loads.

Tools: Torque/Amp-Hour Calculator

I have provided a tool on my website to do these drive train calculations. This is convenient, but it’s well worth the effort to understand how to do the calculations for yourself. To use the calculator go to **http://architeuthis-dux.net/tcr-home.asp** and click on “Tools.” SV

PRODUCT REVIEW — Scorpion ECSs

● by Kevin Berry

Robot Power (**www.robotpower.com**) carries a line of speed controllers popular among bot builders. All Scorpion ESCs have a receiver Battery Eliminator and signal loss fail-safe functions.

The Scorpion XL small/medium robot speed control is a dual-channel 12.5A continuous, 45A peak, 28V dual, fully reversible H-bridge speed control with R/C interface designed for beetle weight and larger combat robots. It includes a weapon and “invert” input, as well as options for on-board mixing or tank-style steering.

The Scorpion HX small robot speed control has a three-channel, 2.5A continuous, 6A peak, 18V dual H-bridge, with 12A weapon control, integrated speed control, and R/C interface designed for ant and beetle weight combat robots. It weighs 22 grams, and has the same weapon, invert, and mixing options as the XL.

“I’ve used my Scorpion Hx for two years with zero problems. It is extremely forgiving to overcurrent. I’ve used it in every iteration of my beetleweight robot “Enemy.” As for toughness, I’ve been blasted to the top of arenas and KO’d so hard that I’ve lost 3/8” titanium axles but the Scorpion never glitched,” states Jim Iocca, NovaRobots, Conshohocken, PA.

The Scorpion Mini small robot speed control is a micro-sized, one channel, 2.5A continuous, 6A peak, 18V H-bridge, with R/C interface designed for ant and beetle weight combat robots. It weighs only 5.5 grams.

“I’ve used the Scorpion Mini in my antweight, Sweet Revenge, for a while now. I bolt them right to the floor, so mounting them is really easy. Their size is pretty good, and they’ve been hit and twisted but no leads have ever broken. I definitely like them,” states Ray Barsa, Team Slackers United, Glen Rock, NJ.


Scorpion Mini


Scorpion XL

RESULTS

Combots — Sony Game Dvelopers Conference, San Jose, CA 3/22/2006 **www.combots.net**. This was a show type format, rather than a competition. Number one ranked HW Sewer Snake from Team Plumbcrazy took on #3 ranked Megabyte, then turned around and fought SHW Super Megabyte, both full body spinners from the Robotic Death Company. Upstate NY Robot Combat Club – Robot Battle IV March 25th 2006 at the Carousel Mall in Syracuse, NY. **www.unyrcc.com** 12 ants and 10 beetles participated, with crowds estimated at 250–300.

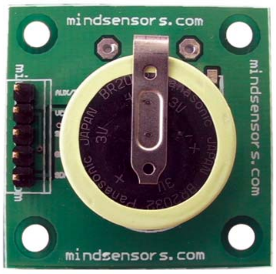
Results: Ants: 1st; Otis, Team Basenji, Lifter. 2nd; Switchblade, Team Sawzall, Eggbeater Vertical Bar Spinner. 3rd; Absolutely Naut VDD, Team Anarchy Robots, Vertical Spinning Blade. Beetles: 1st; BE1, Team Metal Volcano, Undercutter Bar Spinner. 2nd; Buster Blade, Team Basenji, Vertical Spinning Blade. 3rd; Mechanical Menace, Team Steinman, Wedge.


Devil’s Plunger and Swamp Thing 2.

NEW PRODUCTS

ACCESSORIES

I²C Real Time Clock



Mindsensors.com I²C Real Time clock (RTC) enables your robot to keep track of time over longer periods. The RTC tracks time with separate registers for hours, minutes, and seconds. The device also has calendar registers for date, month, year, and day of the week. The calendar is accurate through 2099, with automatic leap year correction.

This RTC can be connected to the host processor using I²C protocol via two wires. RTC utilizes the CMOS technology to achieve ultra low power consumption that is in the 500 nA range. This enables the circuit to be powered by a small lithium cell (part of the product) for more than 10 years.

The processor clock of your robot is perfect for tracking time in fractions of second. However, if you need to track time to the resolution of seconds or above, this I²C RTC is just the product you can use with a fraction of the processor power consumption when compared to equivalent implementation using a robot's processor clock. By using I²C RTC, you can have the robot's processor use its power and time for other high priority tasks. When you need to schedule, processors and peripherals can be put in a sleep mode and the RTC can send an interrupt over I²C to wake up the processor at a scheduled time.

Keeping track of time over a long period is overhead for a processor. It is also affected by the processor load and its crystal frequency variation. The RTC is stabilized for the task of calendar time tracking to the resolution of seconds, and temperature variations do not affect it.

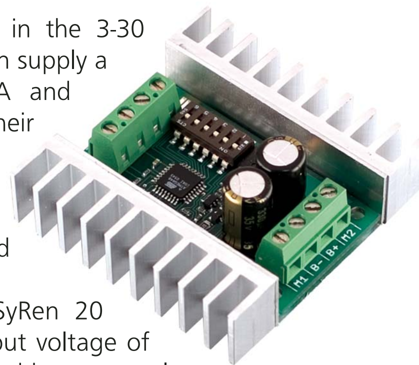
For further information, please contact:

Mindsensors.com
2105 Summerhook Ct.
Glen Allen, VA 23060
Fax: **425 • 984 • 7844**
Email: **info@mindsensors.com**
Website: **www.mindsensors.com**

MOTOR CONTROL

New All-In-One Motor Drivers

SyRen 10 and SyRen 20 are a new pair of all-in-one motor drivers from Dimension Engineering. Aimed at medium-sized robots in the 3-30 pound range, they can supply a DC motor with 10A and 20A continuously. Their peak current ratings are 15A and 30A, respectively. Both ship with preinstalled heatsinks.



SyRen 10 and SyRen 20 both run from an input voltage of 6V to 24V nominal and have a regulated 5V output that can power a radio receiver or microcontroller. They feature a selectable lithium mode that enables the safe use of lithium ion and lithium polymer batteries.

During each switching cycle, SyRen's regenerative topology returns the motor's excess stored inductive energy to the battery. Syren works similarly to a switching DC-DC converter to provide high efficiency across the entire throttle range. This is the first motor driver in its class to do so — conventional designs burn the energy in the motor as heat and are only efficient at full throttle. With a SyRen, motors will run cooler and will get 20-50% longer life when running from batteries. SyRen also recharges the battery when you slow down or reverse — similar to the drive in a hybrid car.

A SyRen can be configured to use an analog voltage, R/C signal pulses, or serial data as the input. All operating modes and options are set using onboard DIP switches, so there are no jumpers to lose or complicated programming instructions to follow. All connections to the SyRen are with screw terminals. This combination means the SyRen can be reused over and over easily.

SyRen features mixing modes in analog and R/C input modes. This allows you to feed a pair of SyRens a forward/back and a left/right signal to control differential drive and tank-style robots without having to resort to computer radios or mixing circuits.

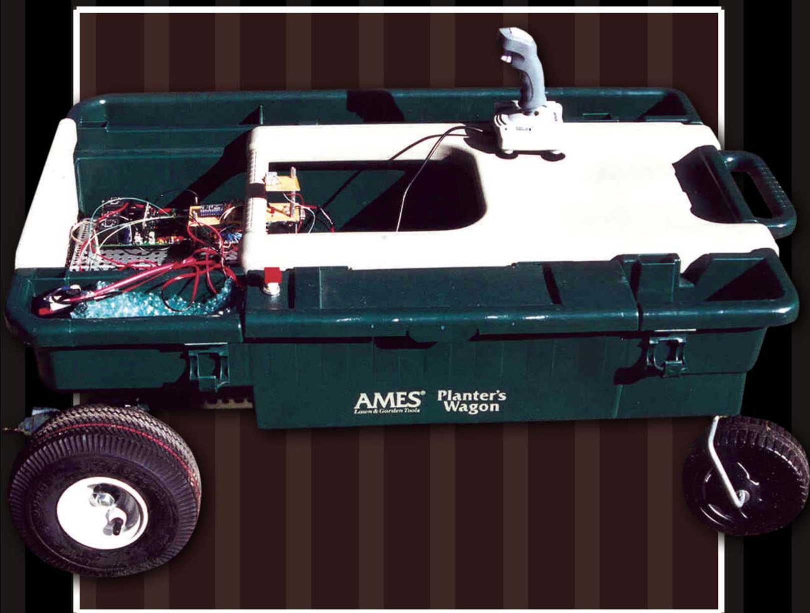
The SyRen series is designed for fast, responsive control so you can make the instant stops and reverses critical in robotics. Thermal and overcurrent protection are built in, so the driver is well protected against damage.

SyRen 10 retails for $49.99 and the SyRen 20 is available for $74.99.

For further information, please contact:

Dimension Engineering
899 Moe Dr. #21
Akron, OH 44310
Website: **www.dimensionengineering.com**

Build an Advanced Two DC Motor Test Platform



by Daniel Ramirez

Electric cars now and in the near future will use electronic power-control techniques to maximize the power output and battery endurance for longer commutes, when space age portable fuel cell technology developed for the International Space Station is commercialized.

IMPORTANT NOTE

All code listings are available on the *SERVO* website at **www.servomagazine.com**

New developments in room-temperature superconductors and nano-tech fibers used for flexible super-conductor wiring are some of the more promising developments that will help make all electric cars a reality. Such vehicles could be a major step to finding a solution to the current dependency on fossil fuels and the increasing negative environmental side effects of their combustion.

Today's cars, especially the new hybrid vehicles include many kinds of electric motors in everything from primary powertrain to simple fans, electric windows, and doors — each with varying power consumption requirements. The more efficiently they work, the better! This is also true for the myriad of electrical consumer items that also use both AC and DC motors. Increasing their efficiency will help reduce our national energy consumption.

Controlling electric motors in new generation hybrid and fully-electric cars is going to pose even greater challenges in power management. Using the dsPIC30F2010's complementary pulse width modulation (CMPWM), quadrature encoder input (QEI), and proportional integral derivative (PID) control and sensing lumps is a relevant baby step to the current and future challenges for bringing a cost-effective and fuel-efficient electric vehicle to market.

Why Build the DC Motor Test Platform?

My goal was to build a low-cost test platform for developing and evalu-

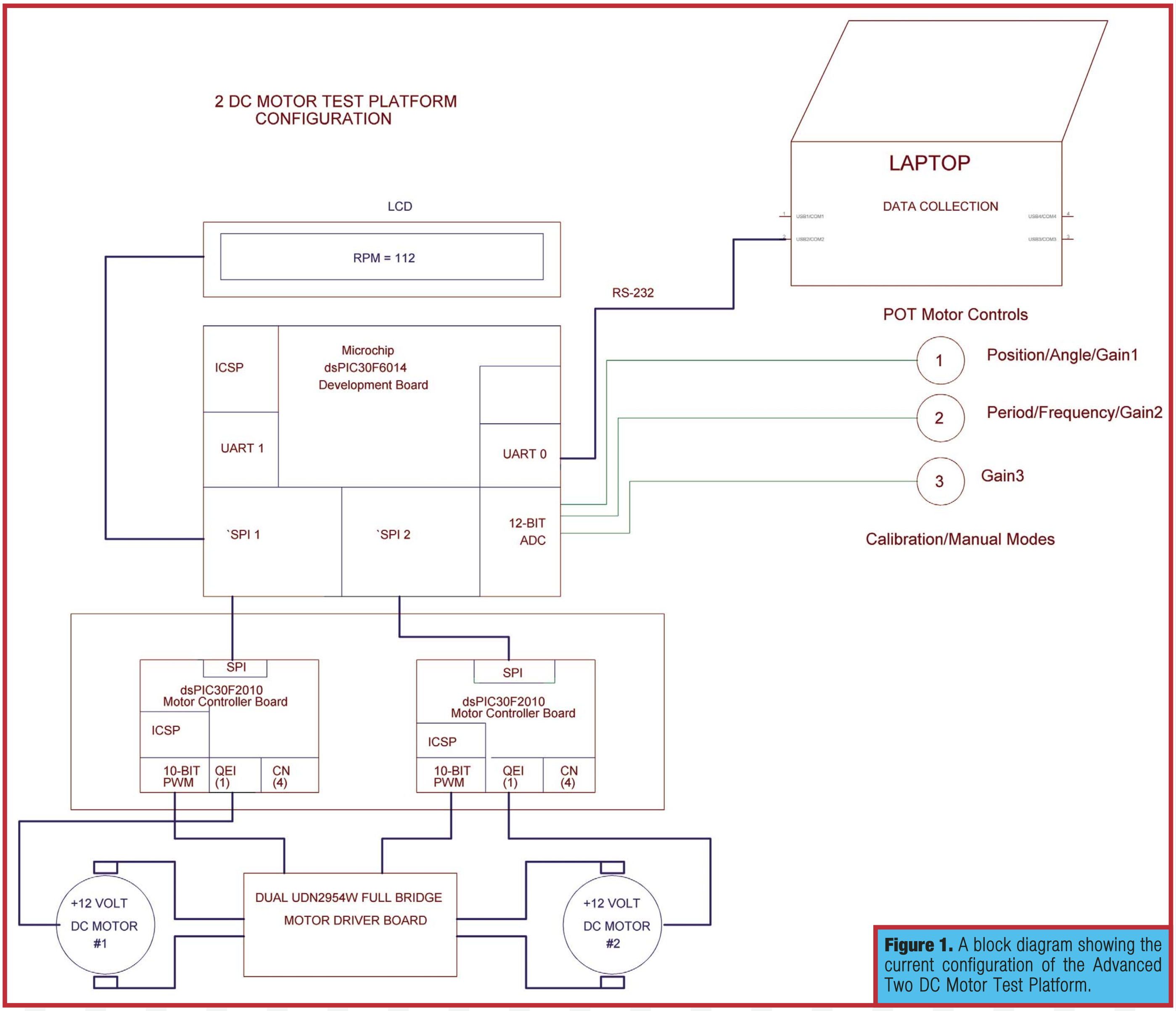


Figure 1. A block diagram showing the current configuration of the Advanced Two DC Motor Test Platform.

ating various PID control algorithms using DC motors for my own robotics applications, some of which are described below.

I am now in the process of actually using the test platform to control two DC motors in both open- and closed-loop modes. My scheme works very well in open-loop, so now it's time to concentrate on closed-loop PID control. Microchip and other vendors provide numerous application notes relating to PID control that I can apply to this application.

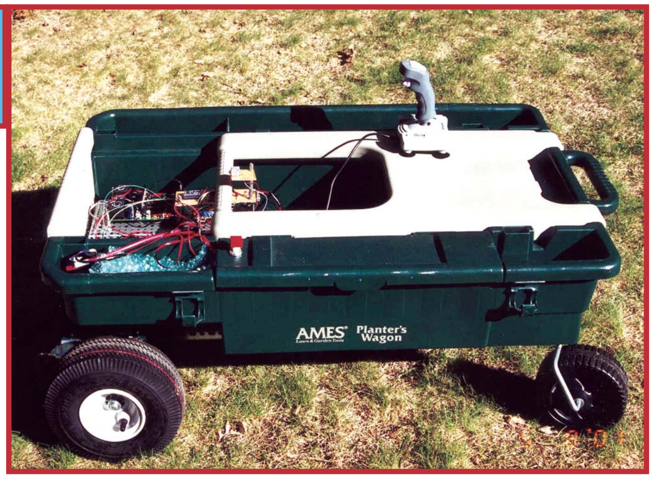
Block Diagram

The system block diagram in Figure 1 shows the various components that comprise the DC Motor Test Platform, including the dsPICDEM™ v1.1 development board used as a master and two F2010 motor controller boards used as peripherals, each connected to the master via the SPI interface for independent control of up to two motors. It should be noted that the F2010 motor controller board has the capability of stand-alone motor control and even limited PID motor control using various application notes provided by Microchip. Although I started out using the F2010 motor controller board as a stand-alone, I have found the complete system more useful because I have been able to run two motors independently.

The 30 MIP F2010 motor controller board in Figure 1 and Photo 1 is the cornerstone to building the motor test platform since it provides the DSP signal processing hardware MAC (multiply-accumulate) and floating point to handle most motor control algorithms. It includes the necessary hardware to generate the PWM signal generation needed to drive the motor amplifier and also provides the QEI necessary to read the position, speed, and direction from the motor's encoder.

In addition, the F2010's timers are used to compute the motor speed (tachometry) and acceleration used with advanced motor-control algorithms, such as PI (proportional integral) and full PID. It also includes one UART for communicating with a host laptop or PC used to display menus and output, and one SPI interface used to communicate with a master SPI controller while acting as a peripheral to it. The SPI interface is crucial to receiving motor commands fast enough for PID control of a DC motor.

Photo 1. The dsPIC30F2010-based DC motor controller board with plenty of room for expansion.



Although the F2010 is able to run simple PID algorithms in a stand-alone mode — as shown in Microchip's application note (AN957) — the master dsPIC30F6014 controller will be needed to handle PID control of two or more motors.

Microchip's future dsPIC33 motor control MCUs, when available, will provide even more memory (128K Flash and 8K RAM) and 40 MIPS performance, and I am looking forward to migrating this application to them as soon as they are available.

The master controller used for PID control is also shown in Figure 1. In this configuration, it is a F6014 based dsPICDEM v1.1 development board that also provides an LCD display along with the necessary hardware resources needed for independent open- and closed-loop control of the motors.

The number of motors may be expanded to four in open-loop mode if additional F2010 MCUs are added using my Motor Messages.

Software PID control of more than two motors at a time may require a faster MCU than the F6014, such as a Philips Power PC, ARM LPC2138, or a TI C40 DSP with fast floating-point support, running a real-time multitasking kernel (VxWorks or Lynx OS).

Additional hardware needed for the test platform in Figure 1 includes one or two dual half or full H-bridge circuits that are paired with each F2010 motor controller. An internal or external +5 volt power supply is needed for the digital electronics and an isolated +6 to +12 volt battery is needed for the motor power supply.

You also have the option to mix-and-match equivalent commercial controllers in order to simplify construction, as long as they match the performance specs used here and are connected in a similar manner.

The dsPIC provides the PWM signal generation necessary to drive the motor amplifier and also provides the QEI necessary to measure the motor's speed, direction, and position.

EarthBot

EarthBot (shown in Photos 2 and 3) is a gardening robot that makes an excellent motor-test platform since it already has two large, geared DC motors for the main drive, and I will soon use it to test control algorithms developed on the smaller motor-test platform described here. EarthBot is currently capable of carrying payloads of up to 175 lbs., so I have plenty of room for the new electronics that will be required along with the laptop-based controller. Other motors that will need to be controlled include a large battery powered electric drill and a powered dirt scoop.

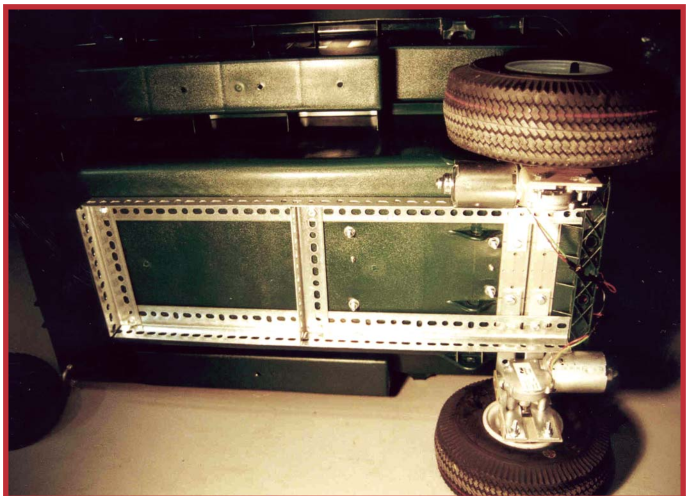


Photo 2. A snapshot of EarthBot on my lawn. It's a great test platform for PID control of up to four motors.

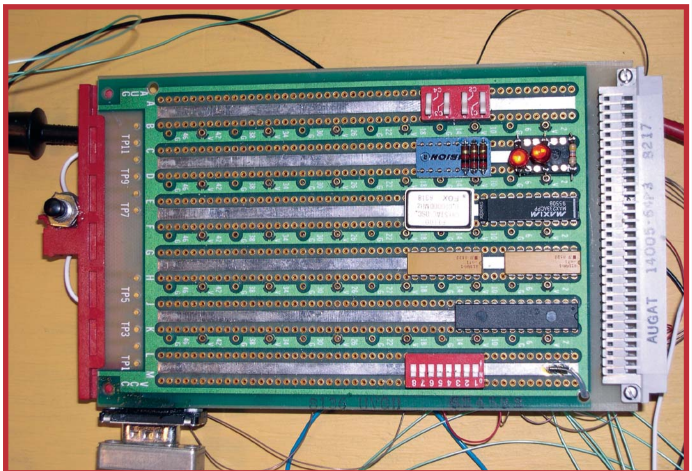


Photo 3. A snapshot of EarthBot's undercarriage showing two very powerful car window geared motors used for propulsion that can easily move me across the lawn using the joystick control.

Presently, I am in the process of upgrading its controller from a Parallax Stamp BS2 microcontroller to new dsPIC-based controllers and a laptop, giving it more intelligence and navigation capabilities. The main reason that I developed the DC motor controller board and test platform was so that I could test and evaluate various motors and control algorithms that will be needed for its new mission to help maintain a garden.

The motor-test platform described here will allow the reader to try out some of these exciting motor control technologies at home, and should be of interest to R/C hobbyists who use the ESC modules to control R/C cars, boats, and airplanes. The AC 3 phase ACIM motor control (AN908) is beyond the scope of this article, but this hardware could be a starting point. Bench testing and PID closed-loop control of up to two heavy-duty DC motors can be done using your PC or laptop as the host, if you want to control the speed or position of a heavy-duty motor using a PID or fuzzy logic closed-loop control algorithm. The serial port could be used to send PWM commands to the DC motor controller's UART if loop speed is not too fast, but higher performance control requires using an SPI interface. Using this DC motor application could get you closer to these goals.

Gearbot

The Gearbot (shown in Photo 4) is another motorized platform that I am building to evaluate and test PID control algorithms and the new motor control hardware. It uses the F2010 motor controller board along with the optical encoder interface, an H-bridge board, and +5 V power supply board to control the two compact and very powerful, geared DC motors with encoders attached. The platform was assembled from Erector Set parts as shown in the photo, but any other construction method (VEX, LEGO, etc.) can be used as long as the motors with encoders are properly fastened.

Photo 4. Gearbot is the perfect test platform for controlling up to two DC motors using PID control.

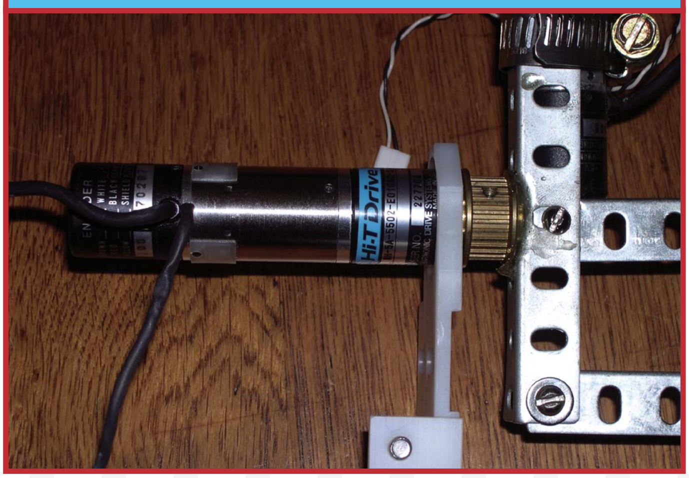
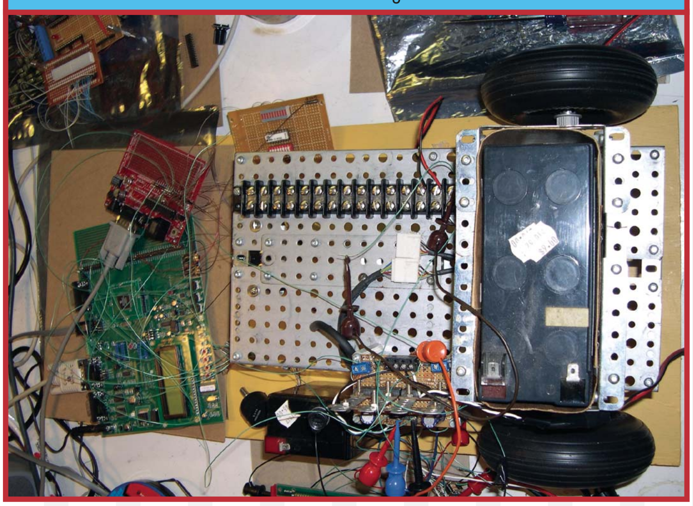


Photo 5. The RH-5A 5502 geared DC motor.



DC Geared Motors

The RH-5A 5502 geared DC motor shown in Photo 5 comes attached with a 100 CPS encoder with an ultra flat gear head and offering zero backlash — features that make these geared motors ideal for mirror-smooth motions required for pointing devices such as security cameras, web cams, and distance measuring devices. They were salvaged from surplus scanner equipment that I found at a local electronics surplus store.

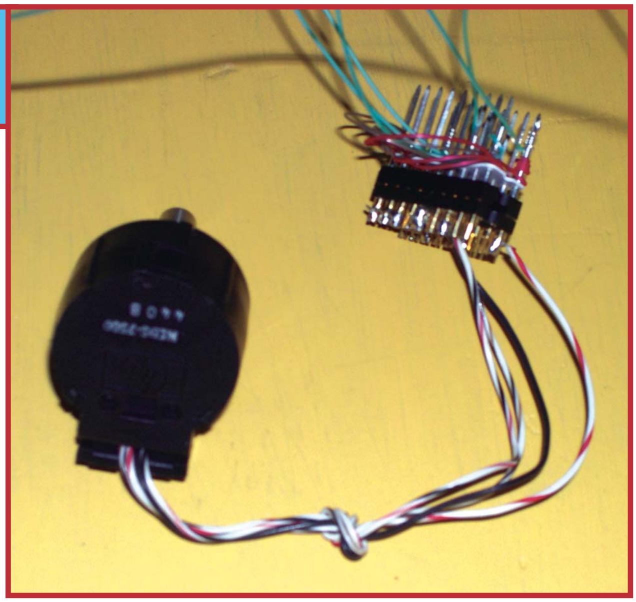
The rated torque for each of the RH-5A 5502 0.29 Nm, the maximum speed of rotation 110 r/min, and the rated speed of rotation 55 r/min. The harmonic drives categorize them as pretty torquey. In fact, they should be able to propel Gearbot so that it will travel over rough terrain with relative ease, even while carrying the heavy battery and necessary electronics. Remember the Jeep-like toy car called Mighty Mike sold in the late 60s? All that was needed to power one was a +6-volt sealed lead acid battery using the DC motor controller and H-bridge boards to control its speed and direction.

Quadrature Encoder Interface

The next step towards obtaining precise motor speed or position control is to use some kind of sensor feedback to give an indication of the motor's actual speed and direction. The sensor usually used for this purpose is called an optical encoder (shown in Photo 6). Other types of encoders exist such as rotary encoders and magnetic Hall-effect encoders. In addition, a variable resistor can provide a resistance or voltage value proportional to the number of turns it has made. These encoders must be connected somehow to the shaft of the motor whose speed and direction is being monitored.

The most common are absolute encoders and relative encoders. The absolute encoders provide the actual position and direction, while relative encoders provide the number of counts and direction.

Photo 6. The HP HEDS-6500 Optical Encoder sensor is used for measuring motor position and speed.



There are many kinds of encoders that are available for measuring speed and position:

- Magnetic encoders (Hall-effect)
- Optical encoders (interrupter+ fine pitch grating wheel)
- Variable resistor or potentiometer
- Mechanical encoder
- Position encoder
- Angular encoder
- Relative encoder
- Absolute encoder
- Resolver

The HP HEDS-6500 encoder in Photo 6 is constructed from an optical disk that provides equal regions of light and dark (grating). Light from an LED will pass through the clear regions and be blocked in the dark regions. Using the encoder's output, you can determine the motor's RPM and direction. This information can be used in a PID algorithm to perform closed-loop control with feedback in order to monitor the motor's speed and to match a set-point speed. The optical encoder acts as the tachometer to measure RPMs and MPH.

It is ideal for synchronizing the speed of two or more dissimilar motors, which helps keep your robot moving in a straight line — even over rough surfaces. This fascinating subject is more advanced and I will not pursue it in this article, other than to mention a couple sources [1][2].

The F2010's QEI hardware is used to filter and count interrupts generated by an optical or Hall-effect encoder attached to the motor's shaft. With this information, we can measure the motor's position (counts or angular position), speed in RPMs, and rotation (clockwise or counter clockwise). The counts and direction flag returned from the QEI hardware help to characterize the motor's performance and provide feedback to a PID control algorithm. Microchip provides a very useful application note (GS002) that explains how to measure position, speed, and direction. It is freely available on their website.

The QEI peripheral provides various filter options. These can be used to reject noise spikes that affect the encoder counts returned from the motor. Because there is only one QEI per F2010, I also provide a CN (change on notice) optical encoder interface that uses an ISR to update the state of the A, B, and Index encoder inputs from a second encoder connected to CN pins. The ISR version does not provide filtering capabilities and may be limited in frequency.

The DC motor controller uses the QEI peripheral driver functions supplied by the dsPIC C compiler library including functions: OpenQEI, ConfigIntQEI, ReadQEI, and CloseQEI. These functions are called to obtain the current position (in counts) and direction from the QEI peripheral. An example of how to use the QEI is shown in Listing 1. The complete

project, qei.c which returns motor encoder counts is included with this article's source code.

Pulse Width Modulation

Automobiles use gearboxes and transmissions as the primary means to step down the speed of the engine. These mechanical means are also available to electric vehicles, but an electric motor has other features that can be leveraged including regenerative braking, coasting, and PWM-based speed control. The gas and air mix flowing to an internal combustion engine control the output RPM and power. PWM works by turning an electric motor on part of the time, and off for the rest of the time. If this on/off switching is done at a frequency higher than what the motor can physically respond to, it is essentially smoothed into a variable speed.

By combining PWM with an H-bridge, you can switch the direction of current flow and also control the motor's direction. Some operational modes in an H-bridge also allow for coasting and braking.

An F2010 provides all the necessary power control and processing hardware to handle almost any motor-control problem. For example, using the six CMPWM outputs and the QEI interface makes it possible to control three-phase AC motors that would make even Nikola Tesla envious. Granted he was 100 years ahead of his time with only the most primitive tools available to him, but it is only recently that new motor designs such as AC and DC brushless motors have been developed and used for R/C car racing, R/C airplanes, and high-torque robotics applications.

The future 40 MIP dsPIC33 family of motor control devices from Microchip will provide even more memory and processing power needed to harness new technologies such as: DC (BLDC) Motor, Sensorless BLDC Control, Sensored BLDC Motor Control, AC induction Motor Control (ACIM), and Vector Control of an ACIM. New sensored brushless DC motors (BLDC) such as the NOVAK HV-MAXX Brushless provide more power for its size than most other DC motors do. Controlling these types of motors requires precise control of the brushless motor's coils in order to obtain maximum performance. The F2010 is ideally suited for this kind of motor control.

The method used to control each DC motor is to invoke the PWM peripheral driver functions supplied by the

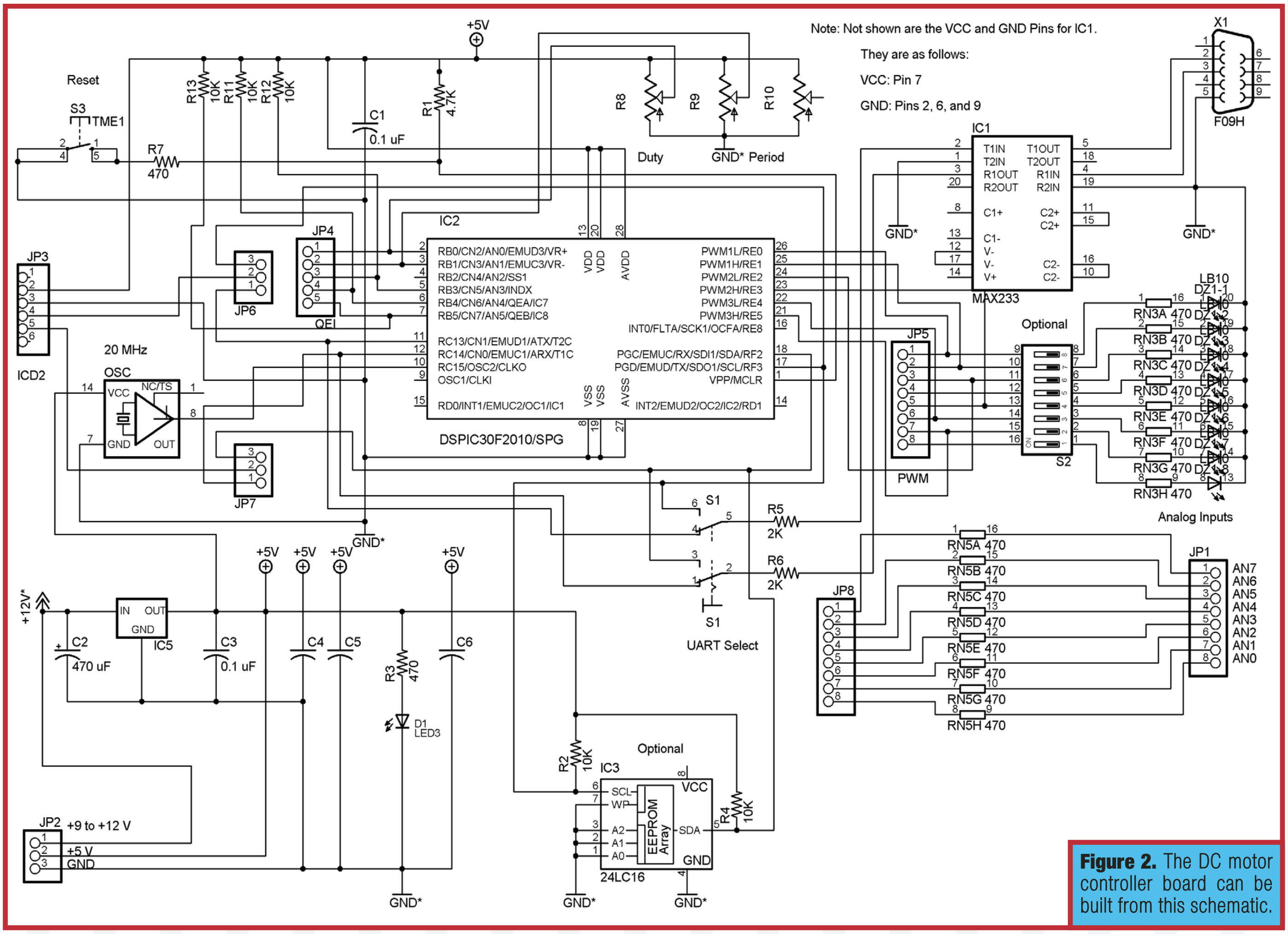


Figure 2. The DC motor controller board can be built from this schematic.

dsPIC C compiler library (pwm.h). I only used one of the CMPWM outputs in this application.

These functions set the PWM frequency and duty cycle and initialize the PWM peripheral hardware registers and port bits, etc.:

- OpenPWM
- SetDC
- SetPeriod
- ClosePWM

Serial Communications Interface (SCI)

To send commands to the motor controller, I used the UART1 available on the F2010 to communicate with a laptop or PC via its serial port. This interface provides a user interface (UI) to display menus and I/O. In addition, I also connected a four-line serial LCD display for a more portable platform that did not require a laptop or PC.

Twin Turbo

The motor controller board can be built from the schematic shown in Figure 2. To use two motors, two identical boards must be built. As you can see from the schematic, I used the F2010 as the DC motor controller with a MAX233 connected to UART 1. In addition, I added a pin header (JP4) to connect to the motor's encoder and provide 10 K pull-ups for channels A, B, Index — as well as +5 V power, and ground to the optical encoder, as shown in the schematic. The 10 K pull-ups wired to JP4 are necessary for encoders that do not provide TTL level outputs. The pull-ups were needed since I could not get the QEI interface to work without them.

To facilitate changing the motor speed and frequency, and to fine-tune the control system, I provided three pots as shown that allow users to vary the PWM duty and PWM period. The third pot is optional and will be used later for changing loop gains during PI and PID fine-tuning. I also used a separate HP HEDS-6500 optical encoder with a knob attached to the shaft to test the QEI.

Full Bridge Motor Driver Boards

The F2010 motor controller board uses the PWM peripheral to

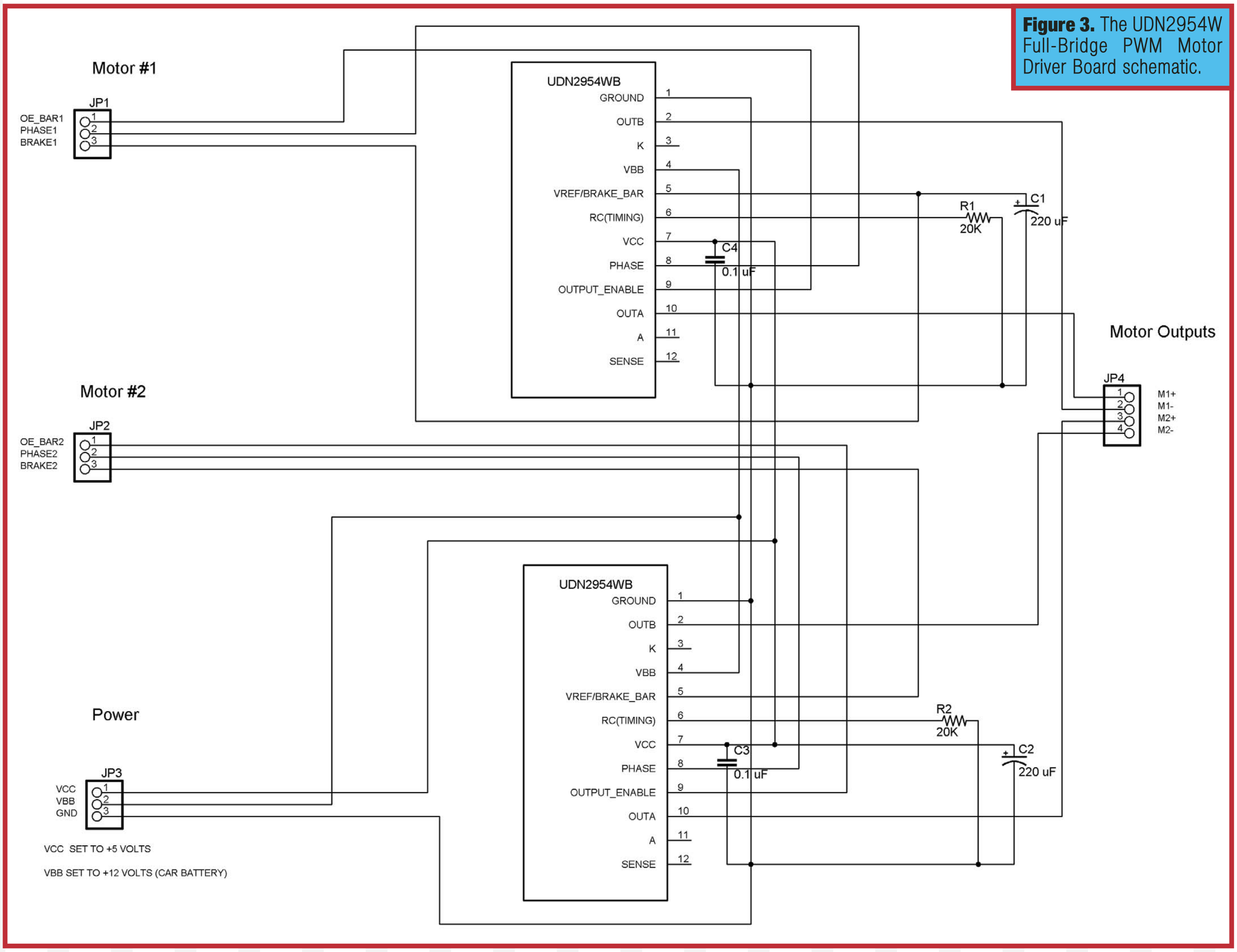


Figure 3. The UDN2954W Full-Bridge PWM Motor Driver Board schematic.

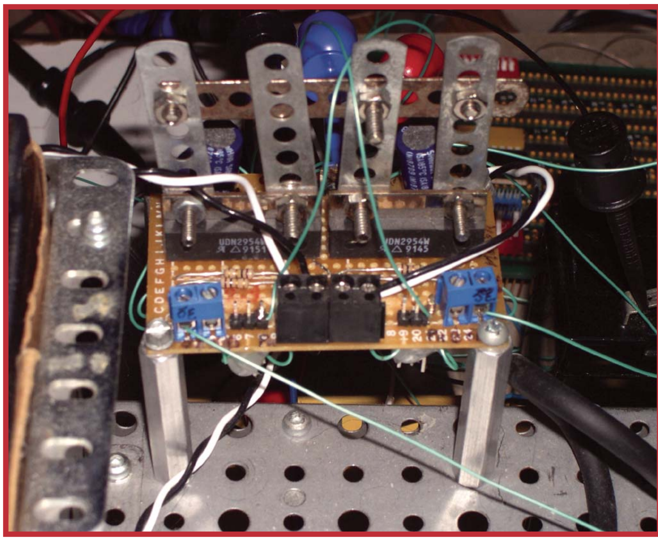


Photo 7. The UDN2954W Full-Bridge PWM Motor Driver Board mounted on Gearbot. Notice the heatsinks made from Erector Set components.

drive one of the DC motors on the Gearbot platform, providing a 10-bit PWM signal to control the duty and frequency of the motor. The motor speed is controlled by varying the PWM signal's pulse width. The PWM output from the F2010 directly drives the dual UDN2954W PWM full bridge motor driver board, shown in Photo 7, that can handle up to 50 V DC and currents as high as 3.5 amps.

Use the H-bridge circuit shown in detail in Figure 3 to build the board. I used heavy gauge wire-wrap although it can be built using standard PCB techniques and components that have "thick" traces with wide pads for power signals that connect directly to the DC motors.

This will insure a very compact DC motor controller. A complete H-bridge circuit is required for each DC motor that will be used to control up to a maximum of two DC motors. The full-bridge circuits may be placed on the same PCB as long as there is room for heatsinks (attached to each UDN2954W full-bridge driver IC). Commercial H-bridge boards of various prices and performance specs are available from ads in this magazine and on the Internet.

Power Supply

The main power supply for all the digital electronics is shown in Figure 4. I directly powered the analog motors using surplus +6V/+12V sealed lead acid batteries. Make sure the analog supply is kept isolated from the digital supply.

Floating Point Support

Microchip dsPIC MCUs support both single and double precision IEEE-567 floats including all the trigonometric and scientific functions. DSP features take advantage of the Harvard architecture and circular addressing features and also support based instructions used for implementing algorithms such as: PID control, digital filters, Kalman filtering, audio processing, etc. (using the dsPIC's new math libraries). The PID algorithm and digital filter implementation take advantage of the dsPIC's lightening fast DSP instructions to support fast single- or double-precision floating point support using the on-chip MAC instructions.

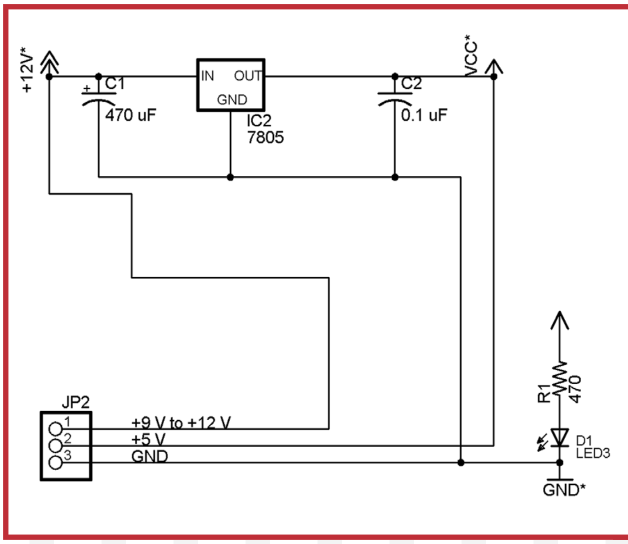


Figure 4. External +5 volt Digital Supply.

Change Notification Inputs

It is possible to control up to two motors without having to add another F2010 to the design. Looking into the extensive data sheets and application notes, I discovered another method for reading the encoders and Hall-effect sensors which involves connecting the A, B, and Index outputs from a second Optical Encoder or set of Hall-effect sensors to the F2010 CN inputs using the method described in application note AN957.

The trade-off for this is added processing in the form of a CN interrupt service routine to read the state of the change on notification bits and decode the associated encoder data.

The advantage to this is that it makes it possible to control a second DC motor without using the QEI interface, so that separate encoder feedback is provided to the control loop for the second DC motor.

Implementing an ISR to capture interrupts generated by the CN pins makes it possible to obtain encoder counts in a similar manner to the QEI peripheral. This means that two encoders can be connected to a single F2010 MCU, thus providing feedback from up to two DC motors with encoders while using two of the complementary PWM outputs to independently send each motor a PWM command from the control loop.

Limitations to this method include added performance overhead for the speed of the motor. The encoder inputs may also require filtering, which is normally handled by the QEI interface filter options.

Although the F2010 is quite capable of doing this filtering, it would require additional code, which leaves less room for communications and PID algorithms.

More capable dsPIC30FXXX devices and the upcoming dsPIC33 MCUs should alleviate this bottleneck with more memory and faster processing. Using these devices could allow control of two DC motors by using CN interrupts with digital filtering for the second encoder.

Motor Commands Messages

In order to allow future growth for my test platform and meet my goal of independently controlling up to two DC motors in both open-loop and closed-loop control modes, I devised motor messages that are common to both the master and slave motor controllers. These messages allow a F6014 master controller to exchange messages between multiple F2010 motor controllers. The C structure of one of these messages is given in Listing 2. The complete message structures that are common to both master and slave applications are located in the dcmotor16.h header file. Other status information such as current motor speed, position, and direction along with current power consumption and temperature can also be reported if sensed and returned in appropriate status messages.

As you'll see from the listing, I provide enough memory to control up to six DC motors independently. These limits may be increased or decreased depending upon the needs of the reader, by adjusting the number of motors and the size of the Rx and Tx SPI communication buffers. Other kinds of messages could be defined in a similar manner to include a status return, heartbeat, etc.

Motor messages sent from the SPI master controller containing PWM motor commands for each motor are distributed to the selected SPI slave DC motor controllers, and status and encoder counts are returned to the master controller during the SPI data exchanges. The SPI interface is fast enough to provide sensor feedback to the PID algorithm running on the master controller while providing multiprocessing to alleviate it from the overhead associated with fast encoder interrupts, digital filtering, and the complementary PWM output which the F2010 motor controller handles efficiently.

To effectively use the DC Motor Controller for your own applications, you must first create messages that can be transmitted over the I2C interface from your master controller to the slave DC motor controller. The messages usually contain the command to be executed by the slave along with any data parameters and a CRC or checksum to check the validity of the transmitted message, since motors are generally noisy and can cause the commands and data to be corrupted by induced spikes from the back-EMF or from sparks generated by the brushes. Although it is possible to remove the simple checksum logic that I added, I don't recommend it because it could cause your robot or device to behave erratically.

Since the F2010 does not have enough memory and peripheral resources (only one QEI channel) to fully implement complex PID or fuzzy logic control algorithms with more than one motor, I decided to use a simple SPI communications network between a master controller and one or two F2010 slave controllers. The second controller is optional, depending on the application.

I defined the following dsPIC C functions shown in Listing 3, for the F6014 master application which are required to build and send the required DC motor control messages for both the SPI master controller and the slave DC motor controller. If a controller other than a dsPIC is used, these messages need to be generated somehow while keeping the byte order the same.

Using this technique with daisy-chained SPI or I2C motor controllers, it is possible to exercise PID control of two or more motors. Just select each one, in turn, via the SPI slave select pins or individual I2C addresses that correspond to each motor. Independent control of each would require multiple instances of PID control loops for each motor.

Motor Test Messages

To test the interfaces to each motor, I devised the communications test demonstrated in Listing 4 that shows how to send simple open-loop PWM period and PWM duty commands to two DC motors and returns the corresponding encoder counts reported from each of the F2010 QEI peripherals.

DC Motor Platform SPI Master Application

The dcmotorm16.c application runs on the F6014 which is onboard the dsPICDEM™ v1.1 development board as shown in the system block diagram. It provides the open-loop control of up to two DC motors while providing encoder feedback using the 16-bit SPI interface and motor messages. I am currently developing PID algorithms to perform closed-loop control of up to two DC motors similar to Rug Warrior described in [1].

In order to set up the master DC motor controller for SPI reception, the MSSP must be configured as shown in Listing 5. The SPI master is required to read any data sent by the SPI slave otherwise the system will hang and data transmission or data reception will cease due to this error state.

DC Motor Platform SPI Slave Application

The F2010 DC motor controller uses the SPI peripheral to act as a 16-bit SPI slave to any host controller that supports the 16-bit SPI interface

as in the system block diagram. It requires an interrupt service routine that handles the incoming SPI bytes sent by the SPI master. These are collected by the ISR given in Listing 6 and stored in a 36-word buffer. As each byte is received, another byte is sent back to the SPI master. The output data contains any status or response information that the SPI master requested.

In order to set up the DC motor controller for SPI reception, the MSSP must be configured as shown in Listing 7. These settings must correspond to those of the SPI master device otherwise it will not work correctly.

A problem that I found when using the Slave Select (SS_BAR) signal was that it would not work unless I placed a 1K ohm resistor in series with SS_BAR when connecting it to the SPI master. I found this solution in a Microchip errata sheet.

I used the SPI software library functions provided by the Microchip dsPIC C compiler by including the "spi.h" header file and making the appropriate calls.

Open Loop Control

The motor-test platform application allows the operator to characterize a motor and obtain values for gains or transfer functions using a simple menu to change the motor frequency, period, and duty required to set the MCPWM registers. In addition, I provide three potentiometers to conveniently change the PWM duty and PWM period (frequency), while also reporting the encoder counts, position, angle, and direction via the serial interface to HyperTerminal or a serial LCD display.

PID Control

PID control is a feedback-based controller that monitors an error in the system (its deviation from some desired value or set point) and makes corrections based on three criteria. The proportional response is based on the magnitude of the observed error, the integral of that error (accumulated over time), and the derivative of the error (the rate at which the error changes over time).

The encoder provides to the control algorithm the actual speed and direction of the motor so that it can compute the error between the commanded motor speed and the actual motor speed, and then compensate for it.

The general equation for PID control includes three terms and is defined as:

c(t) = Pe(t) + Pi * integral(e(t)) dt - Pd*de/dt

where:

c(t) — the control variable — is the correction factor to be applied to the system. In this case, the new PWM motor command sent to the motor.

Pe — the coefficient for the observed error (using QEI for encoder feedback) — is generally based on the error (e) between some user-defined set point (SP) and some measured process variable (PV).

Pi is the coefficient for the integrated error.

Pd is the coefficient for the derivative of the error.

PID Tuning

Tuning of a PID involves the adjustment of Pe, Pi, and Pd gains to achieve some user-defined "optimal"

REFERENCES

BOOKS

[1] Jones, Joseph, L., "Mobile Robots Inspiration to Implementation," AK Peters, Limited.

[2] Martin, Fred, G., "Robotic Explorations: A Hands-On Introduction to Engineering," Prentice Hall, Inc., 2001.

LINKS

- **www.seattlerobotics.org/encoder/200108/using_a_pid.html**
- **www.engin.umich.edu/group/ctm/PID/PID.html**
- **/www.tcnj.edu/~rgraham/PID-tuning.html**
- **www.microchip.com**
- Astro Flight Inc.: **www.astroflight.com**
- Novak Electronics, Inc.: **www.teamnovak.com**
- HD Systems, Inc.: **www.hdsi.net/aboutus/**

MicroChip AppNOTES

- GS002
Measuring Speed and Position with the QEI Module
- AN901
Using the dsPIC30F for Sensorless BLDC Control
27-Jun-2005
- AN908
Using the dsPIC30F for Vector Control of an ACIM
27-Jun-2005
- AN957
Sensored BLDC Motor Control Using d sPIC30F2010
23-Jun-2005
- AN962
Implementing Auto Baud on dsPIC30F Devices
27-Jun-2005
- AN984
An Introduction to AC Induction Motor Control Using the dsPIC30F MCU
27-Jun-2005
- AN992
Sensorless BLDC Motor Control Using dsPIC30F2010
26-Jul-2005

character of system response. It is a subject that can get very involved, especially when dealing with "optimal control" of a plant or motor or cruise control (electric car), or mechanical systems such as the hard disk voice coil used to position the magnetic head. The field of control theory studies the techniques to provide maximum performance from the system being considered. This fascinating field is hard to grasp at first, but there are many readily-available tools and tutorials on the web to help understand it. There are even programs that can auto-tune the system to obtain optimal PID gains.

Upon further investigation, I found out that just having a PID algorithm does not guarantee that you will obtain optimal or even stable control of a system. Fine-tuning a system and obtaining control is an iterative process. There is plenty of fine-tuning and tweaking constants, along with analyzing data collected (Bode plots, root locus plots) that are used to reach the optimal system values. These include minimizing overshoot and maximizing frequency response. Depending on the type of control system (linear vs. non-linear), various mathematical tools such as Laplace transforms, state space, and digital control are used. In fact, new mathematical methods such as fuzzy logic have been used for control of camera focusing *(to un-fuzzify the image? - Ed.)*

The hobbyist may find manual tuning methods good enough but for precise servo motor position or velocity control, the manual methods fall short. It is possible to fine-tune the PID control loop by manually adjusting the Pe, Pi, and Pd gains using the three on-board potentiometers which also double in function for position, speed, and frequency (PWM period) controls during normal operation.

PID Control of Two DC Motors

The necessity for control theory becomes very apparent when you need to fine-tune your control loop to meet your performance goals (overshoot, stability, frequency, etc.).

Microchip provides many PID control algorithms to use with their DC motor evaluation boards. The F2010-related application notes should also work with my Test Platform, while others can be re-hosted from other MCUs to the dsPIC MCUs. It should be noted that these PID algorithms will work best with the motors used in the application notes because the gains have already been fine-tuned. Motor parameters will need to be changed for other DC motors. I have all the related application notes listed in the references sidebar. Both velocity and position feedback are available through optical encoders connected to the CCP capture peripheral or Port B.

Matlab is a tool that provides many control system tools to help you model and generate PI and PID control logic in C. It also allows you to model your application using tools and control-related functions to provide optimal control algorithms systems in electric cars, chemical processes, or robots.

During my research of this subject on the Internet, I encountered two excellent sites that provided me with guidance in this field. The Seattle Robotics Society site had an excellent article written on PID control that explained some easier methods. Carnegie Mellon and the University of Michigan (who support FIRST Robotics competitions) also have excellent PID control tutorials and general control theory tutorials using Matlab. This tutorial uses the Control System toolbox plug-in that is available from Mathworks.

The examples include classic control problems such as balancing a ball on a beam, an upside down pendulum, and servo motor control using root locus, Laplace transform, and state control for each example. Even if you do not have Matlab or the Control Tool Box, the presentation is still worth looking at since it provides diagrams, models, equations, and plots for various kinds of PID control.

Calculus, physics, and control theory books abound in technical libraries. I found a few of these books in my company's technical library, which I use for reference.

Data Collection

The system provides data collection via the SCI to plot the position, velocity, and acceleration graphs for any motor.

This DC motor controller does not provide current sensing or thermal overload protection. It also requires a separate half or full H-bridge circuit, and does not have interrupt support for counting encoder pulses.

Dynamic bench testing of DC motors is accomplished using the dsPIC's QEI peripheral to monitor the motor's position, speed, and direction. The motor speed may be verified by using a commercial strobe light to measure the RPM of a selected motor, as a second check on the speed, and to calibrate the encoder readings.

Firmware

The tools required to build and program the motor controller include the Microchip ICD2, available for around $120. Also needed is the MPLAB IDE, which contains an in-circuit serial programmer (ICSP), a simulator, and an in-circuit debugger (ICD) to flash and debug the applications. You may also want to download the demo dsPIC30 C compiler (good for 60 days), which is needed only if you plan to customize or modify the C source, since I provide the necessary hex files.

The firmware was written in Microchip dsPIC30 C for the F6014 and F2010 using MPLAB, and using the ICD to flash and debug the applications.

Motor Test Bench Applications

The motor test bench application is the firmware from Microchip application note GS002, flashed to a dsPIC30F2010 using the ICD2. It provides you with the means to measure a motor's position or angle (quadrature encoder counts), speed, and direction (forward or backward) using QEI from one of the dsPIC30F2010 motor controller boards.

Open-Loop Control/Operational Test Results

The completed hardware is tested by connecting one or two DC motors with attached encoders to the JP1and JP2 encoder headers provided. First, the motor PWM frequency and direction is specified and PWM commands are sent to the selected motor. It should start running at the selected speed. The motor ID value is required to select the motor under command. When I powered up the hardware, I saw the motors speed up and slow down and stop as I expected. In addition to this, I also sent the DC motor controller messages to change the PWM frequency and messages to return each motor's state. These all worked fine in my application.

Going Further

Have you ever wanted to harness the incredible GHz processing available from today's PCs and laptops, but found the timing issues and interface to your hardware were too much to handle? Another idea that I am exploring for closed-loop PID control of more than two DC motors is to use the USB and SPI interfaces via a PIC18F4550 MCU-based motor controller. I could then use it to connect a PC or laptop to the F2010 motor controller boards and run multiple instances of re-hosted PID closed-loop control using either Windows NT, XP, or Linux using GNAT Ada, C++, or JAVA which support multi-tasking or multi-threading. The data is passed from the PC or laptop to the DC motor controllers via the USB bus using the motor messages. The motor ID is used to select which motor to send commands to and receive encoder counts from.

By learning about these exciting new developments maybe we can bring about a new environmentally-friendly means of transportation that finally gives us independence from the internal combustion engine. Certainly Japanese automakers are going in the right direction with their hybrid cars. SV

by Dave Calkins

ComBots Cup and RFL National Championship

Since the cancellation of both BattleBots and Robot Wars on TV, several competitions have started across the globe to host combat robot events. Builders and fans alike fell in love with the sport, and the simple cancellation of a TV show would not stop them from continuing to watch or compete in such a beloved and popular sport.

In late 2002, most of the event organizers and robot builders in America banded together to form the Robot Fighting League, led by Steve Brown (founder of Steel Conflict), Zander Rose (builder of champions Toro and T-Minus) and a few others. The goal of the RFL was to have a common rule set so that combat robot builders could always count on all events being run essentially the same way, with robots always holding to the same specifications. Identical weight classes, common safety rules, and published weapon regulations all were standardized for the RFL rule set, and bylaws also allowed for new competitions to become part of the RFL.

Shortly after its formation, Steve Brown (first commissioner of the RFL) decided that there should be a pinnacle to the events that were happening around the country — and the "Triangle Series Championship" was formed. The next year, it was renamed the RFL National Championship, and the event now crowns the best robots in nine different weight classes — from a third of a pound up to 340 pounds.

Goal of the ComBots Cup

Due to many factors, there were fewer than normal large-scale events in 2005, so only a very few robots in the 120-, 220-, and 340-pound classes were qualified to compete at the Nationals. Unfortunately, builders with unqualified large bots still wanted to compete, and it's the large robots that audience members wish to see. The audience is the key to long-term growth of the sport — without big robots, the audience appeal will be minimal, and without an audience, event organizers cannot continue to put on events. So, in addition to hosting the RFL National Championship Invitational, ComBots also decided to host an Open event so more robots could compete in the three big weight classes (120, 220, and 340 pounds.)

The goal of the ComBots Cup was three-fold: attract new contestants; reward current builders; and get retired teams to come back. The key was in offering the largest prize in the history of the RFL — $10,000. *It worked*. Five times as many robots registered for the ComBots Cup Open as had been qualified for the RFL Nationals.

While the total number of robots involved in combat robotics continues to grow each year, talented veterans still retire, which is a great loss for the sport. Building the larger, more crowd-pleasing robots also holds a fairly high entry cost for newcomers — a good heavyweight robot can cost between $3,000 and $20,000. Yet, veteran builders who hadn't competed in years came out of retirement for the challenge. Builders such as Carlo Bertocchini returned with his champion robot Biohazard, while Donald Hutson of Tazbot and Diesector fame built a new bot — Karkas 2. Several currently competing teams built new robots or fixed up old bots, while entirely new teams also took up the challenge.

Over 100 robots across all nine weight classes came for the dual-challenge event. Some teams who had qualified for the RFL National Championship even chose to fight their robots in both the National Championship fight, as well as in the ComBots Cup (more fights means both more fun and greater glory, you know.)

High-Tech Creations

Many people who have only seen combat robot competitions on television have no idea just how complex the robots are, nor how exciting the matches can be.

Combots are not just large-scale R/C cars. The vast majority have custom electronics and mechanics worthy of NASCAR. Robots using pneumatics usually have custom-built electronic regulators with multiple PICs. Many teams design and build their own speed controllers, far better than those marketed. Almost all robots have custom machined parts (CNC or hand-milled) to make their creations a reality. And all of them are engineering wonders.

There were many great fights over the course of the weekend. One of the earliest was also one of the most anticipated. After having most of its armor ripped off and its drive train significantly damaged in an early match, returning BattleBots champion Biohazard was scheduled for the last fight of the evening

> **“Many people who have only seen combat robot competitions on television have no idea just how complex the robots are, nor how exciting the matches can be.”**


Photo 1

Photo 5


Photo 2


Photo 6


Photo 3

PHOTO 1. The half-time show included some rather unusual android fights.

PHOTO 2. Winning a match is no piece of cake ...

PHOTO 3. White hot feathers of sparks fly off Megabyte and Brutality in one of their cataclysmic crashes.

PHOTO 4. Donald Hutson and “Karkas” stare down the competition.

PHOTO 5. Ziggy gives Starhawk 3.0 flying lessons.

PHOTO 6. If you want an ice-cold freshly blended smoothy, first you have to go through a heat-exchanger ...


Photo 4

against 18-year-old Paul Ventimiglia and his brand new robot — Brutality.

Paul had fallen in love with combat robots watching BattleBots on TV, and Carlo Bertocchini was his hero. And here, in his second match of the tournament, he was slotted to go up against his hero and his number-one ranked Heavyweight robot. It was a late match, not starting until after 11pm. Everyone was tired, and the audience had gone home. A few builders remained — most to work on their robots, but some to watch this, the last match of the day.

Biohazard had new titanium skirts surrounding its super low profile aluminum body and powerful lifting arm. Biohazard is one of the most effective robots in history at both flipping over opponents and being able to avoid their blows. But Paul had learned many things from watching Carlo and the other veterans, and had also built a very low profile bot with a spinning weapon that covered the full body of his bot. But as the match began, it still seemed likely that Carlo and Biohazard would be able to keep under Brutality's spinning arm of death.

The two robots danced without much damage to either, but just as the first minute of the match ticked away, Paul managed to drive Brutality's front wedge under the champion, lifting him up and into his arm of death. White-hot titanium sparks flew everywhere, and Biohazard lost its skirt faster than a pole dancer on frat night. Within 10 seconds of the first hits, Bertocchini tapped out and Brutality was declared the winner by knockout! In one of the most touching scenes in the history of the sport, the first thing Ventimiglia did — before the robots had even been removed from the arena — was to call his father (three time-zones away) to tell him the news! The young apprentice had usurped the master, and yet still thought to tell his family!

Unfortunately, Ventimiglia would fall later on, unable to make it past the quarterfinals. The finals were left to two dominating champions: Donald Hutson of Team Mutant Robots with Karkas 2, and Matt and Wendy Maxham of Team Plumb Crazy with Sewer Snake. The two robots had met earlier, with Sewer Snake knocking out Karkas 2 in the first minute of the match. This being a double elimination tournament, Karkas 2 had fought back up the ladder to meet Sewer Snake again. This final round was one of the best, hardest hitting, and closest matches in the sport's history.

Both robots have extremely powerful, high-torque frames, both drivers are considered the best in the business, and both have re-configurable weapons systems. The audience shrieked as the ComBox was smashed again and again by these 220-pound behemoths. Both robots proved worthy of the $10,000 prize, and both fought like demons. It was as close to a tie as you could imagine, with either robot edging out the other every other 10 seconds. Only in the last three seconds did Sewer Snake manage to get Karkas 2 up on the rails, helping tipping the judges decision, which even then was split — one judge voting for Karkas 2, the other two voting for Sewer Snake, with a final score of 16 to 17. As close to a tie as possible!

The 100-pound, three-foot trophy and $10,000 check was brought into the arena to the screaming approval of the audience, and the tears of the married couple who had spent so much time and effort to create and perfect. Matt and Wendy held each other as the announcement was made, while Donald Hutson graciously accepted the second place trophy.

Team Plumb Crazy was the big winner overall for the weekend — finishing first in the heavyweight division in both the ComBots Cup and the RFL Nationals with Sewer Snake (fighting in two ladders!), both first and second in the middleweight division of the RFL Nationals with Devil's Plunger and Angry Asp, and second in the lightweight division with Wipe Out #2. Sewer Snake now holds the 2005 Triple Crown championship, having won the ComBots Cup, RFL US Nationals, and the International ROBOlympics (at which it beat not only the best of the American competitors, but the reigning European champion Typhoon 2, as well.)

Combat Robot Events in 2006

If you're looking to build a combat robot, your options are fairly wide, with several events being held throughout 2006. One-third-, one-, and three-pound events happen fairly often in most major cities, and events for robots up to 340 pounds are planned in Louisiana, Texas, California, Minnesota, Wisconsin, Florida, Saskatchewan, and even as far away as Australia, Brazil, and England. The ComBots Cup will once again offer $10,000 (or more) in late 2006. The RFL website (**www.botleague.org**) has a complete schedule of all upcoming events. ROBOlympics (June 16th-18th in San Francisco, CA) will sponsor combat robot events, along with sumo robot events, soccer robots, artbots, the *SERVO*-sponsored 2006 Tetsujin event, and many other robotics competitions (**www.robolympics.net**) SV

COMBOT CUP CHAMPIONS

220 lbs/100 kg ($10,000 prize)
1st: Sewer Snake (Matt Maxham)
2nd: Karkas (Donald Hutson)

340 lbs/154.5 kg
1st: Shovelhead (Doug Groves)
2nd: Super Megabyte (John Mladenik)

120 lbs/54.5 kg
1st: Stewie (Greg Gibson)
2nd: Who's Your Daddy (Jim Yeh)

RFL NATIONAL CHAMPIONS

220 lbs/100 kg
1st: Sewer Snake (Matt Maxham)
2nd: Megabyte (Carl Lewis)

340 lbs/154.5 kg
1st: Shovelhead (Doug Groves)
2nd: Super Megabyte (John Mladenik)

120 lbs/54.5 kg
1st: Devil's Plunger (Matt Maxham)
2nd: Angry Asp (James Arluck)

60 lbs/27.3 kg
1st: Joe 2.0 (Andrew Peterson)
2nd: Wipe Out #2 (Wendy Maxham)

30 lbs/13.6 kg
1st: Totally Offensive (Alex Ueki)
2nd: Killabyte (John Mladenik)

12 lbs/5.5 kg
1st: Li'l Shocker (James Arluck)
2nd: Bullet (Andrew Hebert)

3 lbs/1.4 kg
1st: Buster Blade (Jon Durand)
2nd: 3PD (Andrew Peterson)

1 lb/454g
1st: MC Pee Pants (Andy Sauro)
2nd: Switchblade (Justin Mathews)

5.3 oz/150g
1st: VD (Andy Sauro)
2nd: VDF (Bradley Hanstad)

ComBots Cup & RFL Nat'l Championship


Photo 7


Photo 8


Photo 9

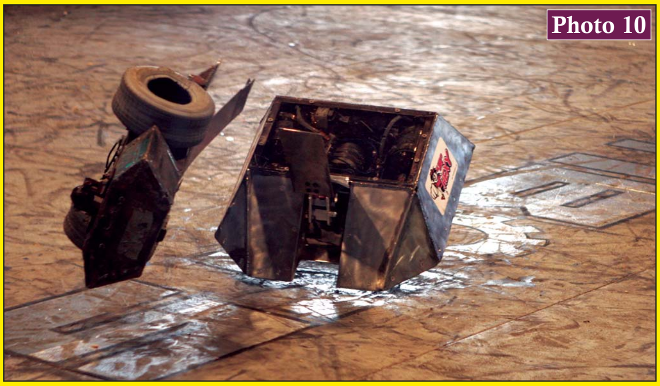


Photo 10


Photo 12


Photo 11

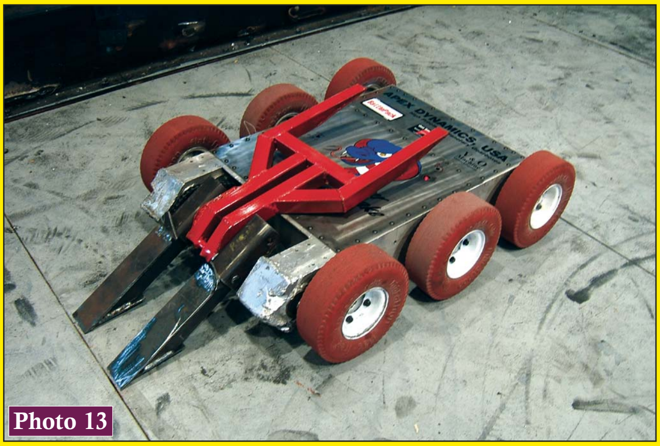


Photo 13

Photo 14

PHOTO 7. The ComBots Cup — a three-foot tall, 120 pound, full scale trophy with a robotic arm reaching out from molten steel — inspired by the "Lady in the Lake."

PHOTO 8. Matt and Wendy Maxham hug and shed tears on being declared the first ever winners of the $10,000 ComBots Cup annual challenge.

PHOTO 9. That's no flying saucer — that's Megabyte hovering above the arena amidst a shower of sparks.

PHOTO 10. Hexy Jr. and Agsma both catch air in a middleweight smash-up.

PHOTO 11. Team Plump Crazy makes some frantic fixes to their middleweight Wipe-Out, seconds before a match.

PHOTO 12. StarHawk 3.0 gives as good as he gets, sending parts of Ziggy flying.

PHOTO 13. Sewer Snake is the highest-ranking heavyweight bot in history.

PHOTO 14. Teammates Marc Demers and Michael Worry confer just before the start of match.

This month we will present the new Robonova-1 from Hitec. The Robonova-1 is a new "edutainment" robotics kit, and it can be acquired in the fully assembled "ready to walk" condition or, as we prefer it, in pieces. The Robonova comes with 16 servo motors, a plethora of prefabricated aluminum body pieces anodized in a slick copper tone, a comprehensive instruction manual, a complete software suite including RoboBASIC V2.5 and Roboscript, a serial cable for programming, an infrared remote control that would be easy to confuse with the one for your TV, and a battery charger.

The Robonova is billed as an "edutainment" robot, and while any robot that can do somersaults is bound to be entertaining, it does not come with any type of canned curriculum like some educational kits do. But how can a bipedal servo walker not teach you something?

Gulliver in a Land of Tiny Screws

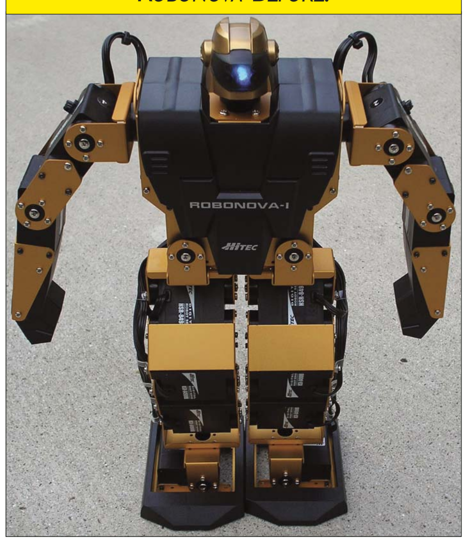
Building a walking robot from scratch is sure to be an educational experience, even as just a way to see what goes into making a bipedal servo walker. In the case of the Robonova, that's mostly a great number of very tiny screws.

Assembling the Robonova was a bit like playing Dr. Frankenstein. The robot was assembled in a piecemeal fashion; a leg here, an arm there, and finally the body. The instructions were unclear and frustrating at times, but the physical assembly was not terribly vexing. Each servo had a conveniently labeled clock of sorts on the driven servo horn to ensure that each limb would have a full range of motion without tricky guess work and estimations on behalf of the builder as to the location of the center of rotation.

The trouble with the instructions was an infectious strain of inconsistency — some pictures would block out the servos (which were conveniently numbered) for a particular limb in a certain order, but then another picture of the actual assembled piece would label them in a different order (not so convenient). We eventually came to the conclusion that some of the pictures were simply not meant to be taken literally, but

THE ROBONOVA KIT.

ROBONOVA BEFORE.



THE PROGRAMMING.

they were a hassle nonetheless.

While a nimble mind could sort out the troublesome instructions, sometimes even the nimblest fingers could have trouble with the fasteners of choice for the Robonova — micro mini screws. The screws, dubbed PH/M and PH/T by the kit and distinguishable only by the coarseness of the threads, could sometimes be so squirrelly to handle that perhaps the best recommendation to any builder would be to lay off of the nail clippers for a few days before attempting to assemble the Robonova. And make sure you have a tiny Phillips head screwdriver ...

Once all of the screws are securely in place (over 100!), the Robonova is ready to be wired up. Thankfully, the instructions become a bit clearer at this point, and the kit includes plastic wire clamps to make routing the wires from 16 servos not quite as tedious a task as it might seem to be. Once all 16 wires are connected to their respective pins on the Robonova's compact circuit board, all that is left to do is attach the body panels. Looking at the mess of wires protruding from the Robonova's back, it's really quite amazing that the back panel fits on, but fit it does.

The Robonova does not come with any preprogrammed modes that allow it to amaze you with its acrobatic ability immediately after construction, but the software CD does come with a cornucopia of template programs ready for downloading. The program that caught our eye was one that allowed the remote control (or Roboremocon, as it's called in the kit) to command the Robonova with a variety of preprogrammed moves, ranging to the most functional task of walking to a very entertaining ballet move where the bot balances on one leg and flaps its arms around as if to imitate a bird.

Of course, the part about installing the infrared sensor to make the Robonova responsive to the remote control only comes after the point where you're told to close the bot up, but it's really no trouble at all to pop it back open and hook up one more wire. One caveat about the infrared sensor, though: the instructions encourage mounting the sensor to the top of the Robonova's head, and while this is the logical place to put the receiver to ensure that you get a clear shot with the remote control, it does seem like a dangerous place for a delicate sensor on a robot that does somersaults and handstands. A protective visor was put on the list of modifications.

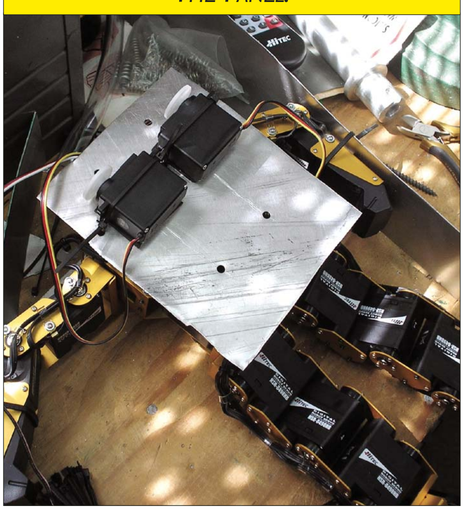
For any last aesthetic touches (not that this slick looking bot necessarily needs any), a full suite of decals are included so you can give your own special touch to your creation.

RoboBASIC Basics

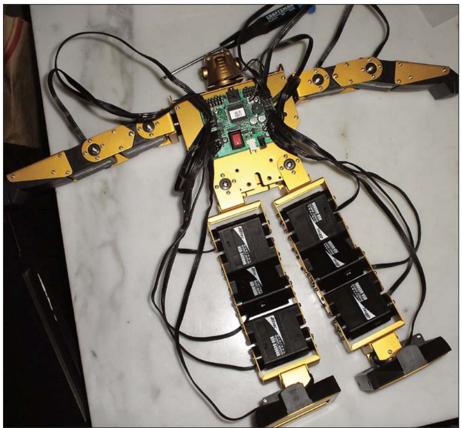
The Robonova comes with its very own programming language, Robo BASIC. RoboBASIC is really very much like any other form of Basic, but just with some special commands thrown in specifically tailored for the Robonova.

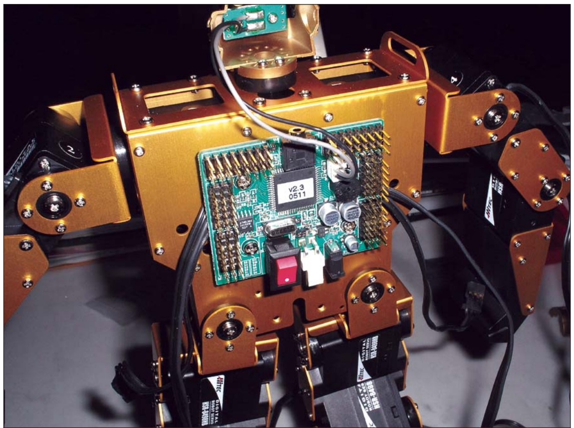
The software CD that comes with

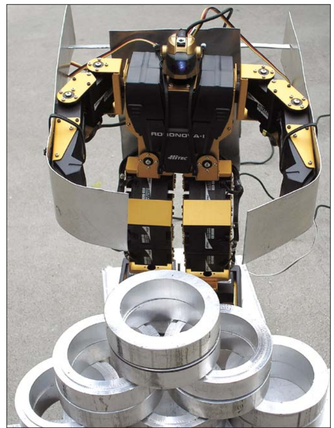
THE PANEL.



SHOWIN' A LITTLE LEG.


ROBONOVA WIRING.


THE ELECTRICAL SETUP.


CYLINDER STACKING.

the Robonova includes a detailed manual on programming in RoboBASIC. The manual includes a brief overview of each command, the syntax of the command, and an example of how the command is used. The manual is logically ordered, starting off with major topics like servo control and then branching out later to more periphery matters like gyros that could be added onto the stock Robonova. The organization of the programming manual even struck as curriculum-like in a sense, or at least there was the potential there to turn the manual into an effective teaching tool to fulfill the "edu" half of "edutainment."

The only possible shortcoming of the programming guide is the lack of comprehensive examples to illustrate the implementation of the various commands in the context of a complete program, but other than that, even programming novices should be well equipped to start coding after leafing or clicking through the manual.

Another helpful software tool is RoboScript. RoboScript is an application that allows for a visual manipulation of the servos for programming — gauges representing each servo are displayed, and adjustments can be made by clicking and dragging. The great part is that once you have crafted a position with this graphical user interface of sorts, the actual RoboBASIC code will be written for you and inserted into your program. It's a great tool for anyone apprehensive about diving straight into RoboBASIC or for someone trying to get over a phobia of analog gauges.

Other cool software tools are some applications that read information directly from the Robonova. After reading the settings of the servos, you can manipulate them one by one on a type of computer waldo that allows the programmer to see exactly what value on a servo corresponds to what position clicking on the dial on the computer picture of the Robonova will cause a corresponding move in the actual robot. This is a great way to take the guesswork out of choreographing motions for the Robonova. A similar tool allows you to reset the zero point of the servos, in case the Robonova ever gets a little off balance in its neutral position.

Tipping the Scales

The bipedal form of the Robonova is already so diverse in its abilities that it was a challenge to think of a way to augment those abilities. Really, what is there to do to improve upon a robot that can already walk (forwards and backwards), do somersaults, execute karate moves, balance on one foot, and do a passable impression of a graceless servo-powered bird?

Leafing through a *SERVO Magazine*, it's actually quite easy to find a way to augment the motion and abilities of the human form. The Tetsujin competition is all about augmenting human motion and abilities with exosuits, so why couldn't we design a mini exosuit for the humanoid Robonova?

For those of you not familiar with the competition (if this is your first issue of *SERVO Magazine*, or if you just fell off the turnip cart yesterday), Tetsujin 2006 will be comprised of three separate challenges: the walking race, weight lifting, and cylinder stacking. These challenges are designed to test everything from strength to mobility to dexterity, and all of them involve the difficult task of designing a mechanical unit to fit around the human body. While increased strength and mobility can easily be achieved for a mechanical unit like the Robonova by just doing something like giving it stronger servos or overvoltaging the bot, a bipedal servo walker could possibly serve as an effective scale model to test designs for an exosuit.

And so we had our challenge. Scale down the Tetsujin challenges and create a mechanical exoskeleton for the Robonova. The first step that we took was to make a comparison between the height and weight of the Robonova and a human being. The height of the robot was easy to measure — about 12.375 inches. The weight was made a little difficult to measure by the fact that we didn't have a decent scale at Robot Central (our garage at home).

Thankfully we go to UCSD, a top research institution with an excellent engineering school, so we were bound to find a scale somewhere. We eventually found a triple beam balance in one of the physics labs and we subsequently found the weight of the

Robonova to be 1.3327 kilograms, or 2.938101 pounds, since we like to use USCS units. That means the Robonova is about a foot tall and three pounds, so using some very rough estimation that means the Robonova is ten times lighter than a person of comparable height.

Prince Myshkin

We were ready to start hacking into the Robonova to equip it with a mechanical exoskeleton, but a mysterious problem hindered our progress. Whenever the Robonova is turned on, it assumes the neutral standing position, but after a few days of operating normally, the Robonova would kick out its left leg defiantly, rendering it unable to stand. Thankfully the software comes with a way to reset the zero point on the servos, but the Robonova's leg eventually kicked out so far that the zero point couldn't even be reset to the point where the Robonova stood upright again. Sometimes the Robonova would even fall victim to inexplicable seizures of uncontrollable motion that earned it the name Prince Myshkin. We were baffled by this problem and eventually settled on the less than ideal solution of reattaching the left hip servo turned slightly backwards so that when the leg kicked out, it actually allowed the Robonova to stand straight up. This made it so the Robonova no longer had the full range of motion in its left hip, but it was worth it to get the bot back its good posture.

Turning the Tables

The first challenge that we would take on with the Robonova would be the cylinder stacking task, a test of dexterity. We're all about giving credit where it's due, and the exosuit design that we'll use for the Robonova is actually a solution for the cylinder stacking task that our dad came up with. The design basically comes in two parts: a turntable to allow for safe turning while carrying a heavy load and arm attachments to help in the lifting itself.

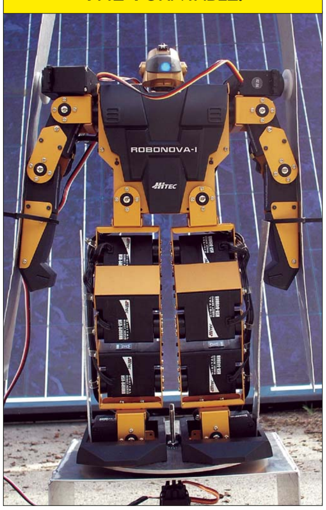
This is a design that was meant for the actual size challenge, but it is actually nicely suited to the Robonova, as well. The idea behind the turntable is to allow safe, ergonomically correct turning. When you're carrying a heavy load (like a 70 pound concrete cylinder), it's very unsafe to turn in the waist because of the heavy load. The turntable would do the dangerous turning instead, allowing the operator to focus on the heavy lifting.

The idea was that the operator would actually stand on top of a large turntable of sorts to achieve this safe turning, and the increased height of the operator might also come in handy while stacking the cylinders into a tall pyramid. This design also works nicely for the Robonova since it can't turn in the waist, and shuffling its feet to turn while holding a scaled down heavy cylinder is iffy at best, certainly prohibitively dangerous when applying such a maneuver to human motion.

An actual size turntable would likely use some powerful motor with crazy gearing, but we didn't have any crazy-strong, crazy-tiny motors for the Robonova's turntable. What worked out nicely for hacking into the Robonova, though, was that the Robonova's board had plenty of open ports for hacking. There were plenty of servos in the garage available for hacking as well, so we picked out a few choice pieces from former FIRST Edurobot kits and proceeded with those. One of our initial ideas with the Robonova's exosuit was that the suit might require a separate brain, but by wiring the extra servos directly to the Robonova, the suit was very much an extension of its "operator" (keeping in the spirit of the Tetsujin mission).

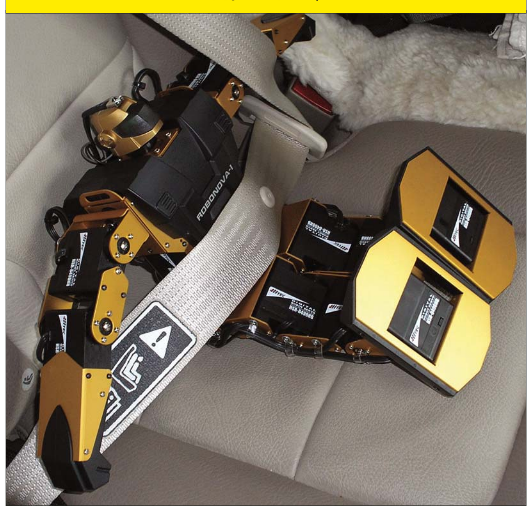
There was plenty of scrap aluminum to craft a turntable out of in Robot Central, so after mashing together a servo from the Edurobot, gears from the VEX Robotics Design System, and discarded aluminum from Cosworth, Inc., we had a working turntable for the Robonova's exosuit. Some possible future modifications include physically bracing the Robonova onto its turntable to help with the torque created by the heavy cylinders, but first we had to get some more of the suit done so we would

THE TURNTABLE.



GLAMOUR SHOT!

ROAD TRIP!



even have torque from a lifted cylinder to worry about in the first place.

If You're Happy and You Know It

The second half of the design would involve strength augmenting arm extensions to help with the actual heavy lifting involved in the cylinder stacking challenge. The idea for the actual size design would be to have arm extensions extend from beyond the hands of the operator to beyond the shoulders of the operator. The extension beyond the hands of the operator would do the actual lifting of the cylinder — this eliminates the risk of any pinching of fingers by allowing the suit (not the driver) to hold on to the cylinders.

The extension beyond the shoulders of the operator would connect to the mechanism that drives the arms and hopefully make them stronger than the operator would be normally. The idea for the actual size design would be to have some kind of large pneumatic or hydraulic piston to power the arm extensions. The idea is to have the piston mounted horizontally between the arm extensions, so that when the piston extends the arms close, and when the piston retracts, the arms open. Not only is this motion clean, simple, and appropriate to the task, but it made the Robonova look like it was clapping when it was not picking up cylinders. Like a toy monkey with cymbals, but a robotic monkey with augmented abilities.

Anyway, we didn't have a pneumatic cylinder small enough for the Robonova, so we decided to use two servos, once again generously donated by the Edurobot, in place of it. While this certainly wouldn't be as strong as high pressure pneumatics, the Robonova was merely a model for the design anyway, and according to our high school physics teacher Vera Korchak — "there are no right or wrong models, just more accurate and less accurate ones." Our design might be less accurate in modeling the actual size design that inspired it, but we were sure that we would still be able to strengthen the Robonova with the mini exosuit. The arm extensions were also particularly useful for the Robonova since the bot had no end effectors besides its plastic fists, and those did not seem well-suited to gripping heavy objects. So the servo powered arm extensions were here to stay.

Of course, lifting a heavy cylinder at the end of the arm extensions could cause a potentially unbalancing torque from the weight of the cylinders, but some more additions to the suit could fix this problem. Some kind of leg braces could keep the operator from tipping over, and some springs of some sort could help in the lifting itself. If there were springs under the arms of the suit, they could compress as the arms are lowered and then help the operator lift the heavy cylinders as they decompressed. Of course, pneumatics or hydraulics might be better suited to an actual size design, but some extra springs from the FIRST kit of parts are perfectly sized for the Robonova.

Numerous tie wraps went into attaching the Robonova's exoskeleton, but other than the fact that the Robonova had to be cut out of its suit, tie wraps seemed to be a good approximation of being strapped into the exosuit. After being strappd into its suit, the Robonova certainly looked stronger ...

Mini Tetsujin, Big Challenge

We know what you're thinking. Oh no, there's so much left to do — testing the cylinder stacking suit and two other challenges altogether, how can they wrap it all up in a few paragraphs? A combination of many factors, including the Robonova's strange erratic behavior, winter quarter finals at UCSD, and the indomitable march of time, fabricating the Robonova's exosuit was simply too ambitious a project to finish all at once. Scaling down the grand challenge of Tetsujin simply cannot be done justice in a single article anyway, so the Super Robonova will be back next time to test its strength, dexterity, and mobility.

Will the Robonova be able to stack heavy cylinders as easily as a magician stacks a house of cards? What exciting designs are there in store for the other two Tetsujin challenges? Will the Robonova develop any other strange afflictions that warrant an allusion to classic Russian literature? All questions and more will be answered in the next installment of Twin Tweaks, so until then, keep tinkering.

To be continued ... **SV**

Programmable Logic

by Gerard Fonte

The goal of this bimonthly column is to provide a basic understanding of the various programmable logic techniques.

There are a lot of powerful low-cost components available today that are rarely considered by hobbyists — and even some engineers — because of unfamiliarity.

You have to be comfortable with the idea and concepts of programmable logic before you will be likely to employ them.

Previously, we looked at ways to implement programmable logic with standard parts. Today, we'll look at small- to medium-sized programmable logic parts. (Later we'll tackle the large parts.) There are a variety of names applied to these small parts: PAL (Programmable Array Logic), GAL (Generic Array Logic), PEEL (Programmable Electrically Erasable Logic) and the generic term PLD (Programmable Logic Device). The mid-sized parts are usually referred to as CPLD (Complex Programmable Logic Device).

PLD Architecture

Figure 1 shows a basic PAL with four inputs and four outputs/inputs. The standard "Invert, AND, OR" circuit described in Part 2 is shown, along with additional output register circuits. Note that the AND gate is shown being wired to every input, inverted input, feedback, and inverted-feedback signal with a "programmable connection." Originally, back in the early 1980's, these connections were made with fusible nickel-chromium links. Nowadays, the fuses are replaced with memory cells that functionally connect or disconnect the AND gate to the appropriate signals.

However, the diagrammatic convention is maintained and simplified to eliminate most of the wires (as shown in the other AND gates). It is the same for the OR gates. Fuses have been replaced with memory cells and the schematic simplified with a single wire to represent a wire to each AND gate. This simplification of the complicated wiring "matrix" is not always apparent or specified in manufacturer's data sheets and can be extremely confusing. There are slight diagram variations between manufacturers, but once the principles are understood, the differences are insignificant.

The general PAL architecture, with dedicated input pins and dual-use output pins, is typical (although there are many variations). There is usually a tri-state option for the outputs (not shown in Figure 1). This option is needed when the pin is used as an input to allow an external signal to drive it without conflict. The output signals are fed back as inputs so that counters and state machines can be created. This feedback makes the device useful for much more than just logic decoding. Once a signal is decoded with the Invert, AND, OR circuit, it is called a "Product Term" or "Min Term" or "Sum of Products" (shown in Figure 1).

The output-register circuit (sometimes called a macro-cell) is present in all but the simplest PALs. It can be just a D-Type flip-flop, or it can be quite complex. There's a lot of variability here. Once a register circuit is present, two additional signals are necessary: a clock signal and a tri-state (or output-enable) signal. (Neither are shown in Figure 1.) Typically, these signals have dedicated pins associated with them. These pins are usually dual-function inputs. That is, if the clock or tri-state functions are not going to be used, they can be employed as additional input pins.

For the basic PALs, the part number can indicate something about its structure. For example, a PAL16L8 has 16 inputs and eight outputs (which are often fed back and can be used as inputs). In the PAL16R8, the "R" indicates there are registers in the output. Many PAL, PEEL, and GAL part numbers follow a similar pattern. However, there are variations. In more modern PLDs, the "R" is often omitted. This is especially true for GAL and other erasable types that can

be substituted for a number of different devices. The clock and output enable lines are not counted. So, if present, they can be used as additional inputs.

One small PAL can replace three to four standard-logic TTL parts and costs a buck or two. The big advantage in using small PALs is the reduction in size and in circuit complexity. Although there is only a small reduction in parts cost, the PCB (Printed Circuit Board) layout can be significantly simplified, and it may be possible to save money here with fewer layers. Additionally, PALs are typically faster than an identical circuit made from discrete TTL parts.

Output Register Circuits

PALs have evolved in tandem with advancements in memory technology. This is because, instead of blowing fuses, memory cells are now used to program these devices. As memory cells become very inexpensive to manufacture, it becomes easy to add more and more complexity to simple PALs. This complexity translates into more functionality for the device.

Figure 2 is an example of an output register circuit that is similar to the one found in the 22V10 device (22 inputs and 10 output/inputs). The Clock and Output Enable pins are directly connected to every flip-flop and output driver of the device. The "programmable" signals used to control the multiplexers (MUX) are set when the device is programmed and cannot be changed (except by re-programming the device — if the device is re-programmable). "Matrix" signals come from other pins or go to the AND inputs via the programmable interconnect matrix. The "min-term" signal comes from the Invert, AND, OR circuit that comprises the core of the PLD.

If you examine Figure 2, you can see the flexibility of this approach. The output multiplexer can select the registered or non-registered min-term, either of which can be inverted. The feedback multiplexer selects either the registered or output value. This allows compound functions to be created that can be very complicated. Remember, for the 22V10 there are 10 of these output register circuits (or macrocells) per chip. This device can replace about five to six typical TTL chips and costs about $4.

After studying the circuits for a

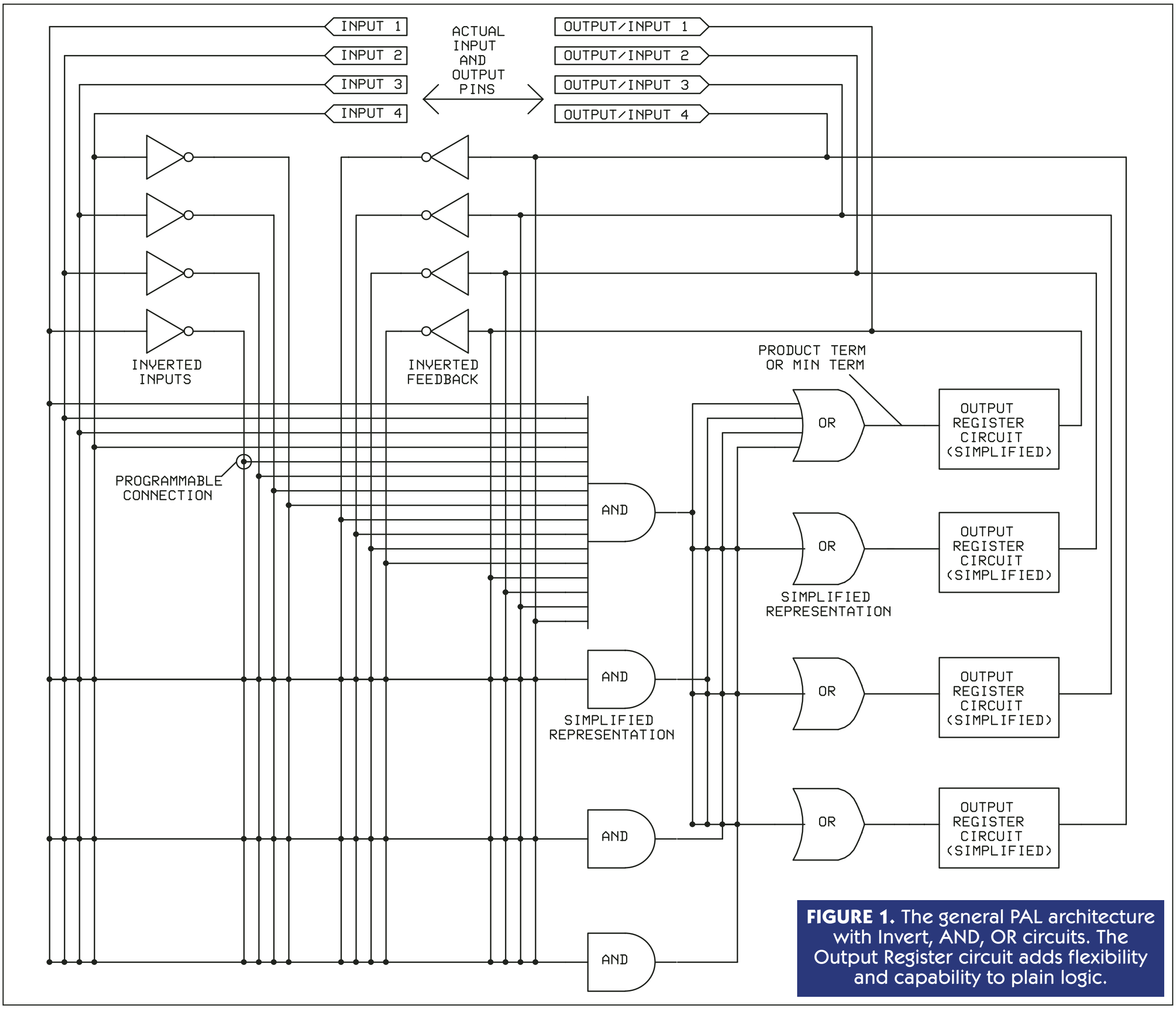


FIGURE 1. The general PAL architecture with Invert, AND, OR circuits. The Output Register circuit adds flexibility and capability to plain logic.

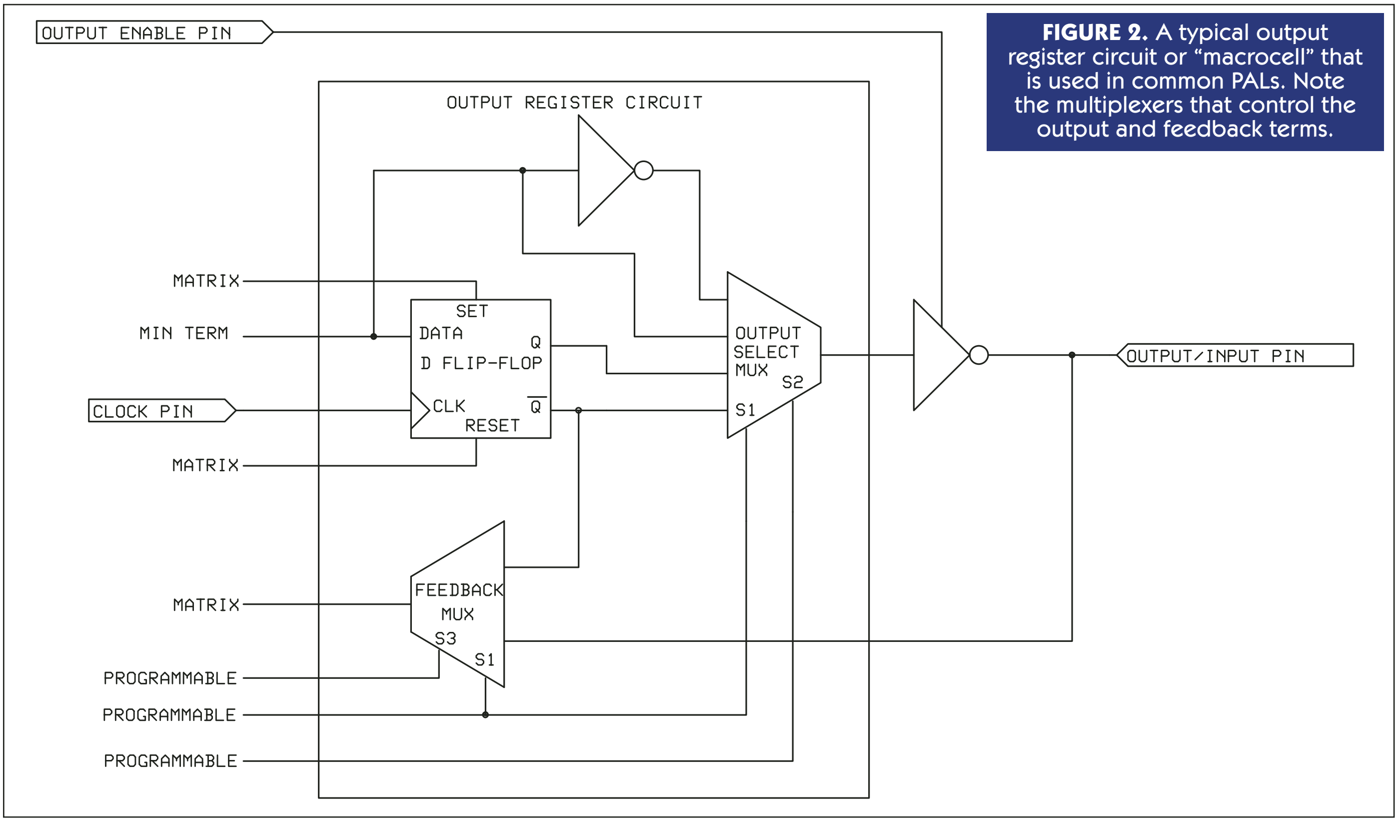


FIGURE 2. A typical output register circuit or "macrocell" that is used in common PALs. Note the multiplexers that control the output and feedback terms.

while you will notice that the OE (Output Enable) line and the Clock line come directly from dedicated input pins. This means that they can't be controlled from a min-term. This initially seems like a significant shortcoming. However, the solution to this problem is to "waste" an output. You create the signal that you want (to control the OE or Clock) and then hard-wire that output to the OE or Clock input. This gives you the flexibility to decode your inputs and use them as desired.

CPLD

CPLDs are the next level in complexity and can be described as being bigger and better. Unfortunately, there does not seem to be any particular standard design or approach. Each manufacturer chooses the method that they think is best. Most often, there are a number of "logic blocks" that are basically equivalent to a large PAL (like the 22V10). These logic blocks are interconnected with another programmable matrix.

Figure 3 shows the general design of a CPLD. In this case, there are four blocks of 16 macrocells each for a total of 64 macrocells. This is over six times more than the 22V10. If you can replace five or six TTL chips with a 22V10, it would seem you should be able to replace about 50 TTL chips with this part. Unfortunately, this isn't the case. Initially, the small PALs were used to replace combinatorial logic (AND, OR gates). But as things got more and more complex, the CPLD began to be used to replace counters and registers. After all, there's only so much you can do with plain gates. This makes the number of flip-flops (or registers) in the CPLD an important factor.

With one register per macrocell, there are 64 registers in the CPLD

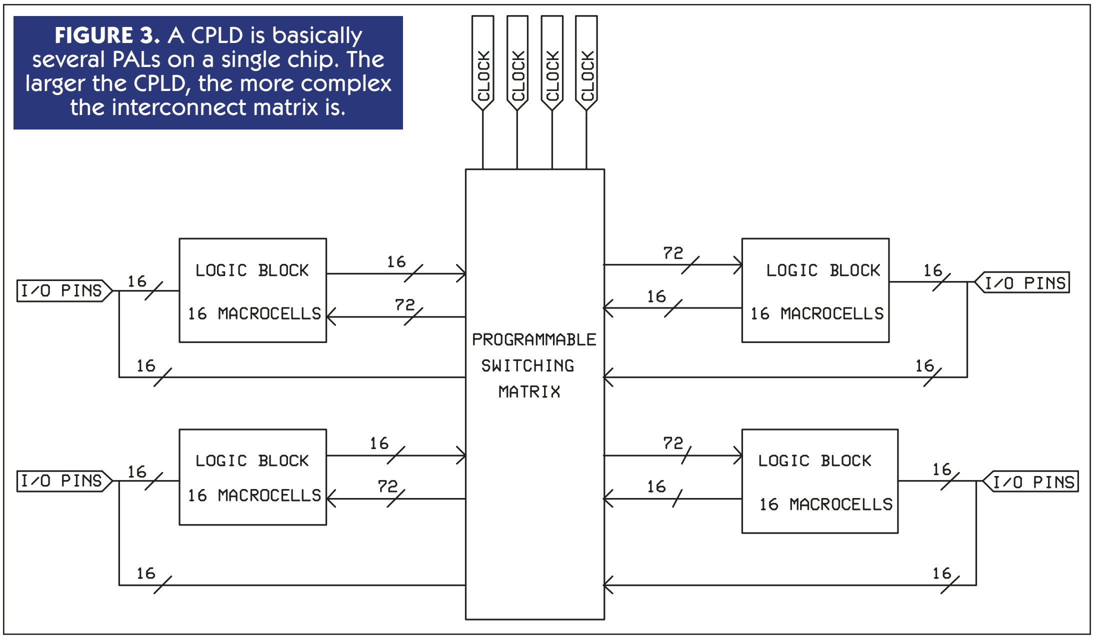


FIGURE 3. A CPLD is basically several PALs on a single chip. The larger the CPLD, the more complex the interconnect matrix is.

shown in Figure 3. (Some CPLDs have hundreds of macrocells/registers.) Suppose your design requires a 16-bit counter. You will need 16 flip-flops to implement that counter — 1/4 of the whole CPLD. The CMOS 4040 IC is a 12-bit counter in one chip that costs about $0.35. This means it can replace only about five of the 4040 chips. That doesn't sound very impressive.

The important point to remember is that the CPLD can replace ANY chips. It makes no sense to use it in place of an IC that already exists. Instead, you'd use a CPLD to replace custom logic. The general term used is "glue logic." For example, say you have a CD ROM drive that you want to interface with a parallel printer port. Making these talk to each other would take a lot of work. But a single CPLD can handle this task. And that is where programmable logic is at its best.

If you want to replace a whole board of TTL parts, you'll need an FPGA (Field Programmable Gate Array) — a type of ASIC (Application Specific Integrated Circuit) — rather than a CPLD. We'll discuss those next time.

Getting Started

There's too much information about specific parts to include here. You'll need to get manufacturer's data from the web or a distributor (see the included list of major players). Often there are free software downloads available, however, some of these files run into the hundreds of megabytes, so downloading them may be impractical. Generally, there are CDs available from distributors that contain this software. Since there are so many different varieties of PALs and CPLDs, you will probably find it useful to specialize using one or two manufacturers' devices rather than trying to learn everybody's proprietary architectures.

Programming these parts requires hardware. Unfortunately, general-purpose programmers are pretty expensive (about $300). They will program most of the small PALs and some of the CPLDs. The larger CPLDs and FPGAs (which we will look at next time) cannot be programmed with these general-purpose programmers. However, the manufacturers often have a "Download Cable" that will program their parts (and only their parts) for $50 to $100. This means that it's probably easier and cheaper to go directly to the CPLD/ASIC applications than to work your way up.

Next time, ASICs. They're not just for millionaires anymore! SV

Resources

Actel Corp.
www.actel.com

Altera Corp
www.altera.com

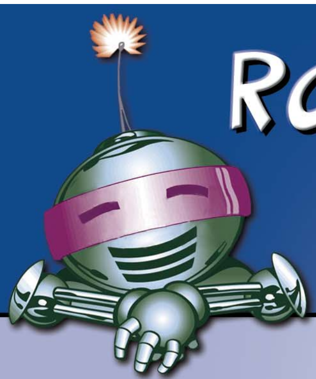
Cypress Semiconductor Corp.
www.cypress.com

Intel Corp.
www.intel.com

Lattice Semiconductor Corp.
www.latticesemi.com

QuickLogic Corp.
www.quicklogic.com

Xilinx Inc.
www.xilinx.com



Robotics Resources

Tune in each month for a heads-up on where to get all of your "robotics resources" for the best prices!

BY GORDON MCCOMB

Heavy Metal

Exploring Robots Made From Aluminum and Light Steel

Today, we live in a plastic world. But this hasn't always been the case. Before the modern dependency on plastic, metal was the mainstay of construction materials. Toys were commonly made of tin. Even into the 1960s, cars were made with metal inside and out, Now, a good portion of automobiles are plastic — and for good reason: it not only reduces the cost of the car, but also the weight. Lighter cars have better gas mileage.

Bucking the trend towards this all-plastic utopia is the amateur and educational robot. Metal robots are more robust than their plastic, wood, or cardboard counterparts. All-metal construction is a near requirement for robots used in combat competitions.

In this installment of Robotics Resources, we'll look at common metals that can be used for robot construction. We'll concentrate mostly on aluminum as this is — by far — the most common metal used in robot construction. We'll also take a quick look at some ready-made metal robots and metal robot construction parts.

Common Metals for Robotics

Metals are routinely broken into classifications: ferrous and non-ferrous. All ferrous metals are made from iron. They're called ferrous because the symbol for iron in the Periodic Table of Elements is *Fe*. Obviously, all metals other than iron are non-ferrous. This includes copper, tin, antimony, aluminum, lead, titanium, zinc, even mercury — which is unique because it melts at room temperature.

Iron is used to make *steel*, which is further classified as having various levels of carbon. The amount of carbon added to the iron during the steel manufacturing process controls the *ductility* of the metal. Ductility is the ease at which the metal is bent, drilled, and cut. High-carbon steel, with 0.6 to 1.5% carbon, is very hard and brittle, and is used to make tools. Parts that require machining are typically made from medium-carbon steel (0.3 to 0.6% carbon). Low-carbon steel (0.05 to 0.3%) — also called mild steel — is used for wrought iron, decorative work, and general household mending chores. It is the type most commonly found at the home improvement store.

Stainless steel is a special formulation of steel with a minimum amount of chromium added in. The chromium develops a microscopic film on the metal that helps it to resist rust and corrosion. Stainless steel is a particularly difficult material to work with, especially if you don't have the proper tools. Though some robot builders opt for stainless as the material of choice — a heavy-duty combat 'bot, for instance — it's not a practical choice for most of us, so we'll leave it out of this month's discussion.

Aluminum is the most common metal used in robot construction projects, part because of cost and part because it is strong yet lightweight. It's also one of the easier metals to cut and drill, and requires only a modest assortment of tools. The aluminum you buy at the hardware store is actually an alloy. The mixture of the alloy varies depending on the desired properties of the finished aluminum material.

Aluminum alloys are identified by number. A common aluminum alloy is 6061, which boasts good machinability — that is, it's not difficult to drill or saw — yet is still lightweight and strong. Other alloys are designed to provide a harder and stronger metal that is able to withstand high heat, but is also much harder to machine. More about aluminum later in this column.

Copper is one of the rare metals that is used in its pure state, though it's also alloyed with other metals. When combined with zinc, copper makes brass; when combined with tin, copper makes bronze. All three are soft metals, and are relatively easy to cut and drill. Copper and its alloys can be readily annealed, which is the process of heating the metal to high (but not melting point) temperatures, then allowing it to cool very slowly. Annealing is used to change such properties as the softness and ductility of the metal. It's commonly used to make springs and so-called springy metal.

Zinc and *tin* are most commonly employed as an alloy ingredient, or as a coating; for example, adding zinc to copper to make brass. It is also used to plate and galvanize steel, a dipping process that coats the steel with a relatively rust-free surface. Tin is a malleable bright metal that is commonly used as a

coating for steel, as tin resists corrosion and inhibits rust. "Tin cans" for food are really made of steel, then coated with tin to protect against rust. They are more accurately "tin-plated steel" cans.

Zinc-plated and *galvanized steel* is common in metal fasteners, as well as various hardware items found at the home improvement stores. What is routinely called an "angle iron" — used to joint two pieces at right angles to one another — is medium-carbon steel that has been galvanized or zinc-plated.

Other metals — some "exotic" and some aren't — also find their way into the robotics workshop. These include nickel, typically used as a plating; titanium, for machining very strong parts; iron, for casting and machining; tungsten carbide; cerro alloys, for casting; and lead, also used for casting.

A Closer Look at Aluminum

As we've noted, aluminum is an alloyed metal, and the percentages of various metals determine the properties of a given aluminum alloy. Table 1 summarizes common aluminum alloys, which are identified by a three- or four-digit number.

Aluminum alloys are also commonly listed with their *temper* rating. For example, aluminum listed as 6063-T52 is alloy 6063, with a temper rating of 52. Tempering is used to change the ductility and other properties of the alloy. The letter designation indicates how the material was tempered:

F — as fabricated
H — strain hardened
T — heat treated
O — annealed
W — solution treated

Alloy Series	Major Alloying Metal
1xxx	Pure aluminum, for smelting
2xxx	Copper
3xxx	Manganese
4xxx	Silicon
5xxx	Magnesium
6xxx	Silicon and magnesium
7xxx	Zinc

TABLE 2.

The first digit of the aluminum alloy indicates its chemical makeup, as shown in Table 2.

Additionally, aluminum bronze alloys (numbered 630, 642, 954, 959) are used for casting or machining replacement parts. You'll find these specialty alloys often used for machining gears, worm drives, and other power transmission components. Other alloys are made for casting rather than machining. Aluminum alloy 319, which has a melting range of 960-1,200° F, is available in ingot form, and is used to cast parts.

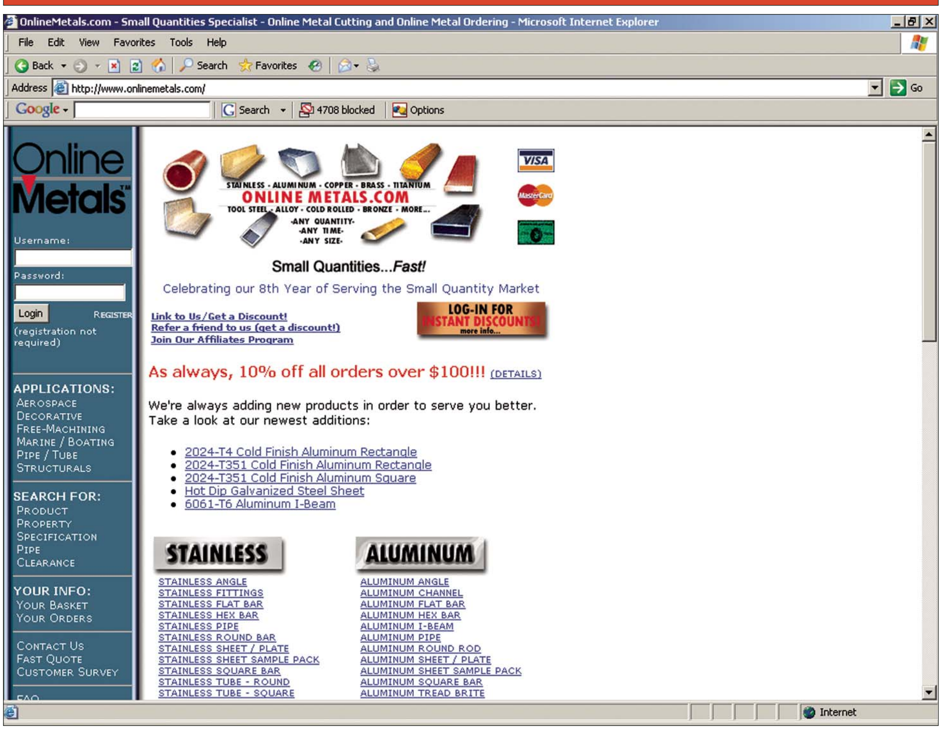
Alloy	Melting Range	Typical Applications
2011	1,005-1,190°F	Machinable parts
2017	955-1,185°F	Machinable parts
2024	935-1,180°F	Fasteners, fittings, wheels
3003	1,190-1,210°F	Sheet metal work, equipment tanks
4032	990-1,060°F	Machinable parts
5052	1,125-1,200°F	Drums, tanks
6013	1,052-1,195°F	Machinable parts
6061	1,080-1,205°F	Pipe fittings, scaffolding
6262	1,080-1,205°F	High temperature engine fittings and parts
7068	890-1,175°F	High temperature engine fittings and parts
7075	890-1,175°F	Gear and other power transmission parts

TABLE 1. Properties of Common Aluminum Alloys.

Except for ingots, most aluminum alloys are available in a wide variety of shapes, including plate, sheet, rod, bar, tube, honeycomb, foil roll, discs, and various extrusions — T-, I-, H-, U- channels, angles, beams, and more. You will find the most basic shapes at the local hardware or home improvement stores; the rest can be found at a well-stocked industrial metals outlet. Check the Yellow Pages for one near you. Additionally, mail order is an option. For example, Online Metals (**www.onlinemetals.com**) offers various aluminum alloys in a variety of shapes.

Aluminum is available as-milled, or anodized. Anodizing protects the

FIGURE 1. Online Metals offers one of the most extensive lines of metal materials, including aluminum, stainless, steel, copper, and more.



aluminum from corrosion. *Anodizing* is commonly found on extruded lengths of angle and channel stock at the hardware store, and is clear (silver) in color, tough black, or gold — other colors are also available from metalworking shops that perform anodizing. Aluminum that has not been anodized is provided *as-milled*. No anodizing is used because the metal is expected to be machined, painted, or anodized by the user.

Though aluminum does not rust, it can corrode if not protected against oxidation. When not anodized, aluminum is usually painted, either with ordinary spray paint, or with *powder coating*. Powder coating is a form of painting where a pigment and a resin are electrostatically charged, then sprayed onto an electrically grounded metal. The result is a uniform coating that is not as susceptible to sagging or dripping, as is the case of aerosol spray painting. Powder coating is primarily the domain of specialty metal coating shops, but there are some consumer-level paint application units you can purchase for small jobs.

Robots of Steel

The word "steel" conjures up images of super strength. The industrial nations of the world have been built on steel — it's what makes possible railroads, skyscrapers, and bridges — and without this material, progress and invention wouldn't be nearly what it is today.

For all the benefits of steel to society, its usefulness as a construction material for robotics is limited by its heavy weight, and its relative difficulty of being drilled, shaped, or cut without the use of heavy-duty metalworking tools — tools most of us don't have. Even so, when you need the strength and rigidity that it provides, steel is hard to beat.

As noted above, steel comes in a variety of forms, dictated by elements added to it during manufacture. Typical of steel for amateur robotics use is the relatively soft low-carbon steel. You can find this material at most well-stocked hardware and home improvement stores, as well as metal supply outlets for the welding trade. In addition to the elemental makeup of the metal, steel varies in how it is heat treated. Heat treating is used to enhance certain physical properties of the metal:

• *Hardening* strengthens the metal, and literally makes it harder. The process also makes the metal more brittle. Hardening is accomplished by heating the metal to specific temperatures, then cooling rapidly. Tools are commonly made of hardened steel. The degree of hardness is determined by the speed at which the metal is cooled, and by its carbon content.

• *Annealing* softens the metal, and makes it more workable. Annealing is accomplished by heating the metal, then allowing it to cool very slowly, sometimes over days. During the cooling process, the grains of the metal can be re-oriented to endow the material with unique properties.

• *Tempering* removes some of the hardness and brittleness of steel, and in doing so makes it tougher. Tempering is often employed after hardening or annealing, in order to achieve unique properties of the steel. Like hardening, the tempering process involves heating the metal to a specific temperature, then rapidly cooling.

• *Case hardening* is a coating process for soft steels, and allows relatively low-carbon steels — such as wrought iron — to be hardened. Methods of case hardening varies, but one such process involves dipping the heated metal in a carbon-rich case hardening compound. The

SUMMARY OF METALS FOR ROBOTICS

Here is a review of metals that are particularly well-suited for the construction of robotics. Each metal is noted with its common use, main benefits, and main drawbacks.

Metal	Common Robotics Applications	Main Benefits	Main Drawbacks
Aluminum	Bases, arms, all structural parts	Reasonably priced, lightweight but strong, easy to cut and drill using proper tools	Plethora of alloys makes picking the right one difficult, can be hard to weld
Brass	Decorative trim, fasteners	Fasteners and trim commonly available at retail stores	Relatively soft, low tensile strength
Bronze	Hose fittings	Readily castable	Fairly heavy
Cerro alloys	Castings	Good alternative to pure lead, low melting points (117-450° F)	Hard to find, can contain toxic metals (e.g. cadmium, lead, and bismuth)
Copper	Fittings	Ductile and readily machined	Expensive, easily tarnishes
Lead	Castings	Very ductile, low melting point, easy to cast	Toxic
Steel	Heavy-duty frames	Very strong, very inexpensive	Rusts if not protected, can be hard to drill and cut
Stainless steel	Heavy-duty frames	Resists rust and corrosion	Some alloys cannot be welded, can be hard to work
Tin	Sheet metal bodies	Soft and malleable, thin sheets ideal for robot bodies, relatively low melting point (450-725° F)	Can be hard to find in various sizes and stocks.
Titanium	Gears, small parts	Extremely light and strong metal	Expensive, very hard to work without the proper cutting/milling tools, high melting point of 3,000° F.

metal is then placed in a furnace to be heated. Only the outer shell of the steel is hardened; the inside core is left soft.

• *Rolling* is a process that extrudes a block of steel into plate, sheet, or other shaped form. During the rolling process, the atoms of the steel are realigned, giving the material added strength. Steel may be cold rolled or hot rolled. Cold rolled steel tends to be thinner yet stronger than hot rolled steel, and is usually more expensive.

Hardening and tempering can be accomplished in the home shop, and is helpful if you're making a robot for heavy-duty chores, or for combat. The most common method of hardening steel is to heat the metal with an oxy-acetylene torch. The torch is "played" over the metal for a uniform heat. After the steel has reached a certain temperature, it is either left to cool in air, or quenched in water, brine, or other liquid. Alas, I can't provide more details here on the ever-exciting topic of hardening metals. There's much more to steelworking than can be discussed here. Consult a good metal-working text on the process.

Ready-Made Metal Robots and Metal Construction Parts

Suppose you like the idea of a metal robot, but don't want to build your own. Understandable, as metal isn't the easiest material to work with, especially if you lack the tools. A number of companies offer metal robot kits and metal construction parts. These kits and parts are pre-cut, pre-bent, and pre-drilled — all you do is fasten them together.

Perhaps the best known metal robot is the BOE-Bot from Parallax (**www.parallax.com**). This flexible little creature is built using a single pre-punched sheet of metal. You complete the kit by attaching motors, wheels, caster, battery holder, and BASIC Stamp 2 microcontroller board. Parallax offers a number of retrokits for the BOE-Bot, which extend its capabilities. One all-metal kit turns the BOE-Bot into a simple hexapod walker; another lets you use Tamiya rubber tank treads, rather than the stock wheels. The company also offers a bipedal robot — the Toddler — made of anodized aluminum.

One of the most notable sources of all-metal robots is CrustCrawler (**www.crustcrawler.com**). The company offers a number of two-, four-, and six-legged kits, as well as a metal arm and gripper kit. As with many metal robot kits available today, all are designed for use with standard model R/C servo motors. Some of the kits are available via the CrustCrawler site, and some are available through Parallax.

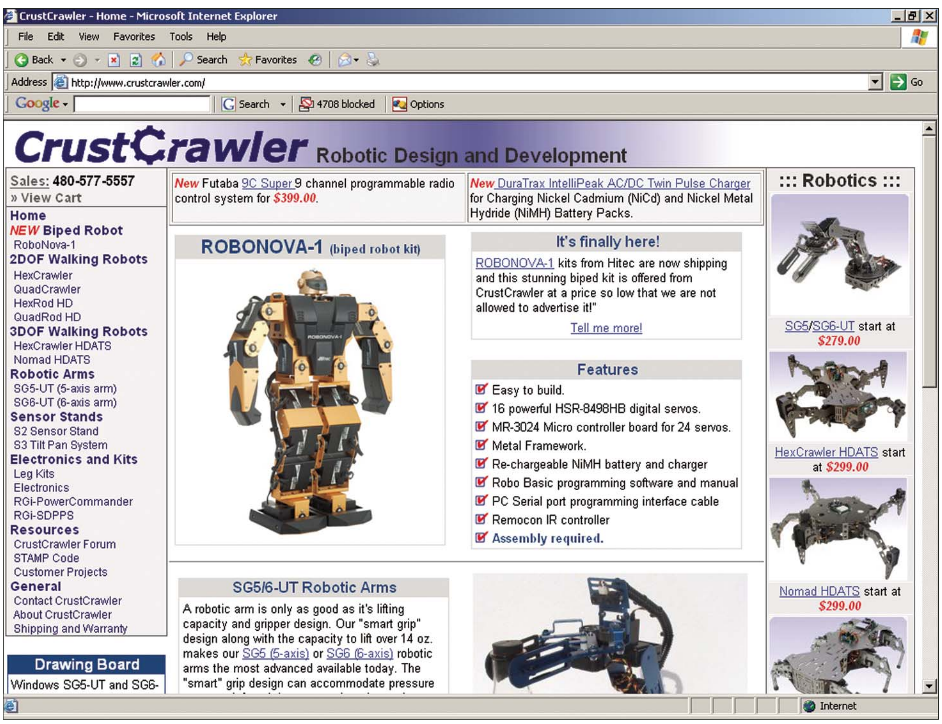
Robotics specialty resource Acroname (**www.acroname.com**) offers several robot kits constructed from sheet metal. There's the PPRK three-wheel robot (which uses a Palm Pilot for its brain), as well as Garcia. The latter you can customize with various finish colors. The Arrick Robotics ARobot metal platform (**www.robotics.com**) is a motorized base designed to be powered by a BASIC Stamp.

If you like the idea of combat robots, but don't want to weld your own, check out Battlekits (**www.battlekits.com**), makers and sellers of small combots. Robot Combat (**www.robotcombat.com**) sells parts and materials for building metal-based fighting robots. For smaller versions, there's Sozbots (**www.sozbots.com**), 16-ounce robots that nevertheless can do quite a bit of damage. Sozbots is also a reseller of the Robo-One walking robot.

Lynxmotion (**www.lynxmotion.com**) sells a set of nifty R/C servo brackets and other parts that allow you to assembly your own metal-based robot. The brackets come in a variety of shapes, which allow you to build all types of robots, including quadrupeds and hexapods, and various types of robotic arms. Along similar lines are the pre-punched metal construction pieces offered by Hobby Engineering (**www.hobbyengineering.com**). The company offers various frame pieces, girders, and brackets.

Robodyssey (**www.robodyssey.com**) offers several of its custom-designed walking and rolling robots cut from lightweight aluminum. All use R/C servo motors for propulsion, and

FIGURE 2. CrustCrawler.com is the creator of numerous high-end all-metal robot its. They also resell notable metal kits, such as the Hitec Robonova.



are designed to interface to a microcontroller. Superdroid Robots (**www.superdroidrobots.com**) manufactures and sells a unique expandable metal robot kit, the SuperDroid Trekker.

For those in the UK, Total Robots (**www.totalrobots.com**) provides several creative kits made entirely or mostly out of metal. They have a radical-looking robot arm, the "Sted-E-Man" bipedal walker, and several rolling bots. Based in Canada, Rogue Robotics (**www.roguerobotics.com**) makes and sells several kits constructed of powder coated aluminum. They offer a multiple-deck rolling robot, as well as a tracked-based kit. The company has resellers worldwide.

We robotics enthusiasts have long used R/C servo motors in our creations. The servo companies have finally caught on, and several now offer kits for building metal bots. Hitec Robotics (**www.hitecrobotics.com**) offers the Robonova, a 12" 16-motor bipedal robot. It's not cheap (about $1,000), but it's guaranteed to get noticed.

The Vex Robotics kits (**www.vexrobotics.com**) contain metal construction pieces, and are reminiscent of the old Erector sets. The kits have been offered through RadioShack (**www.radioshack.com**); check the Vex website for current resellers.

Several companies offer robots for the educational market, and are primarily made of metal. Examples include Dr. Robot (**www.drrobot.com**), the RB5X (**www.rbrobotics.com**), and ActivMedia Robotics (**www.activrobots.com**).

As a general note, the RB5X robot is probably the longest-running robot kit still on the market. It was first introduced in the early 1980s.

ABOUT THE AUTHOR

Gordon McComb is the author of the best-selling Robot Builder's Bonanza, Robot Builder's Sourcebook, and Constructing Robot Bases — all from Tab/McGraw-Hill. In addition to writing books, he operates a small manufacturing company dedicated to low-cost amateur robotics, ***www.budgetrobotics.com*** *He can be reached at robots@robotoid.com*

Metal Sources

The following online sources provide construction metals suitable for use in building amateur robots. Those sources that provide retail sales also offer an online shopping cart; additional online sources (which you can find using Google) offer sales via the traditional online or offline quote system.

In addition to the companies listed here, you'll find most metals for robot building locally. The benefit of obtaining metal from a local source is that you don't have to pay shipping charges; for large and heavy pieces of metal, shipping costs can significantly add to your budget.

• *Hardware stores* carry some aluminum and steel sheets, angle brackets, and rods. Selection is limited, and prices can be high.

• *Hobby stores* sell small aluminum, brass, and copper, in small sheets, rods, tubes, and strips. A common brand sold by stores in North America is K&S Engineering; you can review this company's product line at **www.ksmetals.com**. The products are also available through many online hobby stores, such as Tower Hobbies (**www.towerhobbies.com**).

• *Extruded aluminum* used for machine framing. An example manufacturer is 80/20, Inc. (**www.8020.net**).

• *Metal supply shops* that cater to welders are open to the public. Many sell stock in large pieces, which you can have cut so you can get it home in your car. Tip: Check out the "remnant" bin for odds-and-ends sizes. However, be careful what you buy. If you're looking to weld stainless steel or aluminum, you must be careful to select an alloy that offers good weldability.

• *Restaurant supply stores* — most of which are open to the public — sell many aluminum and stainless materials. Some things to look for include spun bowls, cookie and baking sheets, unusually-shaped utensils, strainers, and even salt and pepper shakers.

Aluminum Association
www.aluminum.org
Trade group for aluminum manufacturers, suppliers, and users.

McMaster-Carr
www.mcmaster.com
Large online materials source; products include various metal plates, rods, tubes, and other shapes.

Metals Depot
www.metalsdepot.com
Web-based retailer of various metal, from plate to tubes, rod, extruded shapes, etc. Convenient cut lengths.

Metric Metal
www.metricmetal.com
Metal construction pieces, in metric sizes only. An $85 minimum order amount applies.

Online Metals
www.onlinemetals.com
Very large variety of metals, including aluminum, brass, and stainless.

Reid Tool and Supply
www.reidtool.com
Good source for extruded aluminum in various "profiles," shapes that are specifically designed for machine framing.

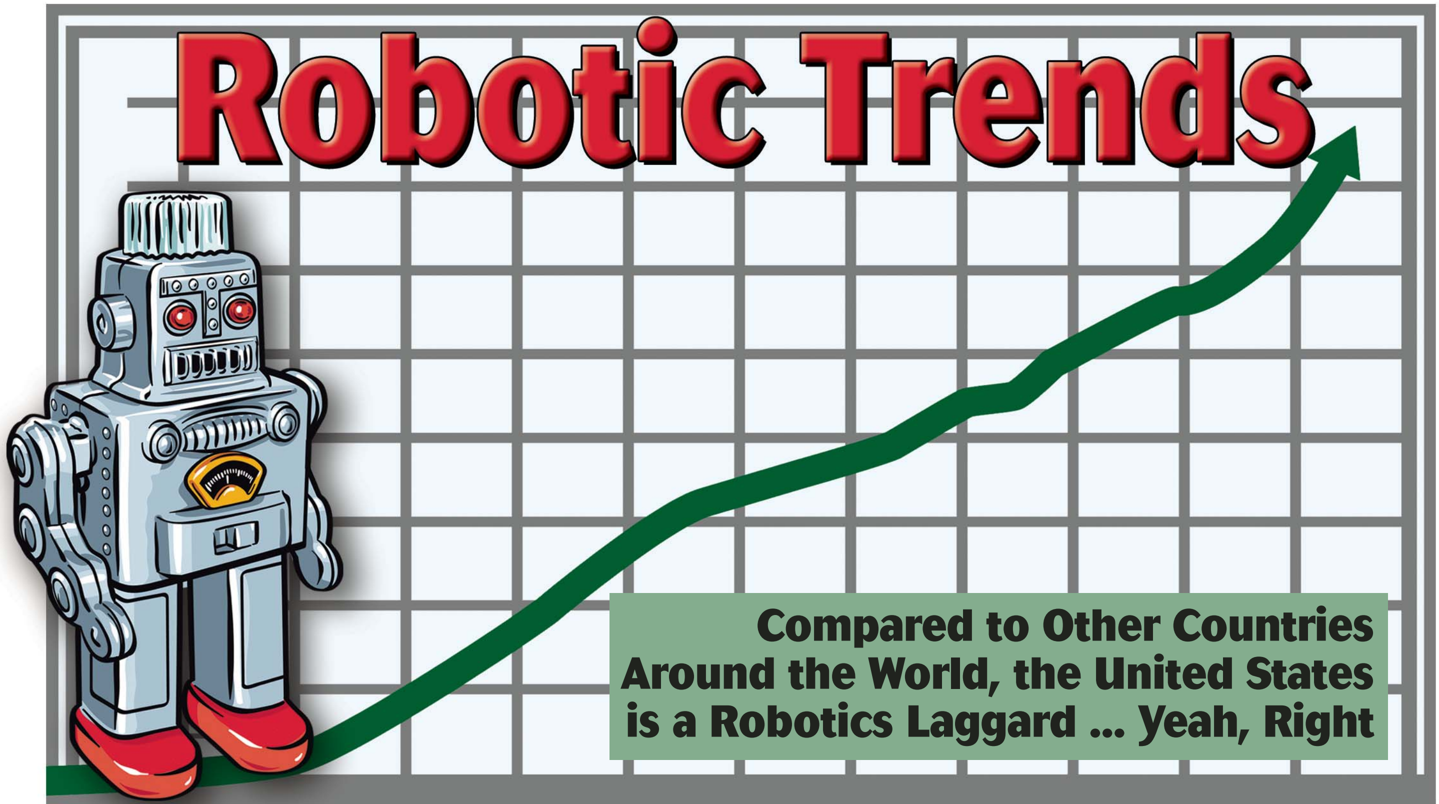
Rigidized Store
www.rigidizedstore.com
Specializes in small quantities of "rigidized" (textured) metal, such as that used for structural plates.

Robix
www.robix.com
Aluminum robot kits.

Small Parts, Inc.
www.smallparts.com
Specialize in small components. Offers online ordering; large inventory.

Titanium Joe
www.titaniumjoe.com
Specializes in titanium sheets, bars, and other shapes. SV

Robotic Trends



Compared to Other Countries Around the World, the United States is a Robotics Laggard ... Yeah, Right

by Dan Kara

Dan Kara is President of Robotics Trends, the producer of the RoboBusiness (www.robobusiness2006.com) and RoboNexus (www.robonexus.com) conferences, and publisher of Robotics Trends (www.roboticstrends.com), an online news, information, and analysis portal covering the personal, service, and mobile robotics market. He can be reached at dk@roboticstrends.com

If you interact with the larger robotics community for a long enough period of time, you cannot help but hear a number of statistics, stories, and 'truisms' that are passed around year after year at business meetings, robotics conferences, and over morning coffee.

I have found that a number of robotics truisms are not true at all, or at least much of the total story is left out. Certain 'facts' only become facts with a capital 'F' with repetition.

A good example of such facts is the commonly held belief that the US lost out to Asia, primarily Japan, on the industrial robotics market because unions were unwilling to let robots into US shops for fear of job loss. While job loss was a concern, it was not the only issue and perhaps not the primary one. Questions relating to capital expenditures on unproven technology (in a recessive period), and the rate and amount of investment return, also came into play. Japanese corporate culture is supportive of long-term investment goals with relatively modest returns on investment. Their US counterparts favor a quicker turn on the buck and a better rate of return. At the time, industrial robotics promised neither of these.

Another one of the robotics truisms is that the US lags Japan, and possibly even Korea in the robotics field. Not quite. The talking points that drive this line of reasoning are typically some combination and permutation of the following:

- **Japan and Korea dominate.**
- **The US lacks a national robotics policy.**
- **There is a lack of interest in technical studies in the US.**
- **The 'new' robotics market is not 'real.'**
- **The US does not have a culture of robotics.**

Japan and Korea Dominate

One of the most popular robotics truisms is that the US lags Japan, and possibly even Korea, in the robotics field. Not quite. The counter argument can be summarized in two words ... military robotics. The US is the leader by far in the research, development, and commercialization of military robotics. Over time, the amount of defense related funding for military robotics initiatives will run into the billions of dollars. More importantly, the spending surge currently underway for DoD-funded robotics research and development is going to pay off in spades for all manner of non-military applications (see Table 1). Commercial field robotics and autonomous transportation are the most obviously beneficiaries, but you can bet that there will be all manner of commercial robotics products that will have benefited directly from DoD flavored research and development initiatives.

Lack of National Robotics Policy

This is largely true for a formal national policy, especially compared to government backed robotics business development initiatives in Korea and Japan. While government support for the robotics industry in Japan is well

> **"For a country without a culture of robotics, competitions and events seem to be extremely popular."**

known, it is important to note that the Koreans are equally dedicated and are backing up their beliefs with cash. For example, the Korean government will spend 316.5 billion won (US $264 million) over the next five years on 16 projects to boost the domestic intelligent robotics industry. A task force of businesses and research groups are seeking ways to link the robotics industry with potential growth industries such as display equipment, next-generation chips, and networked homes, with the stated goal of making Korea one of the world's top three nations in the robotics industry by 2013 (15 percent of the market).

A number of US governmental bodies do serve as funding instruments for robotics research including the National Science Foundation (declining), the National Institute of Standards and Technology (its own activities and small grants to universities), the Department of Energy (nuclear power and hazardous waste), the Department of Health and Human Services (assistive technology, prosthetics), the Department of the Interior (mining, field robotics, underwater robots), and NASA (space exploration and satellite servicing). For many of these groups, however, support for robotics research is flat or declining.

Corporate funding for robotics research is very thin at this time. Some funding for core robotics research is being funneled through universities, but there is little direct investment in robotics research at the corporate level for the development of commercial products. This lack of internal funding for robotics research has been attributed, correctly I believe, to US firm's emphasis on a quick return on investment and an ROI of at least 15%.

What the US lacks in a formal robotics policy, it makes up for with informal efforts beginning with DoD funding of robotics projects. Consider the following:

- For FY 2004-2009, the Army is allocat ing $500 million to unmanned ground platforms;
- DARPA is bankrolling over 40 robot-related projects at universities and private firms;
- The Defense Department is expected to spend up to $10B on unmanned aerial vehicles by the end of the decade; and
- The Navy is spending $50M to develop four prototypes of a surface platform geared for operation in littoral areas and $130M on remote mine hunting systems.

The list could go on and on in terms of the billions of dollars of military funding dedicated to robotics development, to say nothing of homeland security.

Another example of informal US robotics initiatives is the development of three robotics clusters within the US. The New England cluster is anchored by its university system including the Massachusetts Institute of Technology, Boston University, and so on, and is fueled by East Coast venture capitalists in New York and Boston. iRobot, Deka Research, MobileRobots, Foster Miller, and others provide corporate stewardship, as does the Massachusetts Robotics Cluster, a robotics business development group run out of the Massachusetts Technology Leadership Council.

A second robotics cluster, based in Pittsburgh, revolves around Carnegie Mellon University and includes the National Center for Defense Robotics, the National Robotics Engineering Center, the Technology Collaborative (a business development group), and many local robotics firms. San Francisco/Silicon Valley is also a player, but not the dominant one. The Valley is home to Stamford, University of Santa Clara, San Jose Sate, UC Berkley, and a host of other educational powerhouses, as well as many small entrepreneurial robotics firms and traditional computer companies (who are keeping a watchful eye on this new technology sector). West Coast venture capitalist firms (San Francisco, Silicon Valley) are also keenly interested in robotics following the iRobot initial public offering.

NASA	DoD
Improved fire insulation materials	Better bulletproof materials
Better materials for insulation against cold	Night vision goggles
Advances in photography	Long-range sensors
Improvements in solar power technology	Webcam technology
'Coolsuits' for firefighters	Commercial GPS
Satellite communications	Better bulletproof materials
Improved materials for athletic shoes	Night vision goggles
Battery-operated power tools	Long-range sensors

Table 1. Commercial Spin-Off Technology From NASA and DoD Research and Development.

Lack of Interest in Technical Studies

It is true that the number of US college students majoring in computer science is declining. This, however, is largely a function of computers having completely mainstreamed since the late 1970s and thereby losing much of their 'attractiveness of the new.' Let's face it, in the 80s and 90s Information Technology (IT), followed by everything 'Web,' was hot. For today's high school and college students, however, IT is now the equivalent of accounting ... it provides a decent living, but is hardly exciting. The Web continues to spin off new and interesting business models and new versions of the Internet will offer even greater opportunities, and will attract the interest of many prospective engineers. So too with game design and mobile computing, but in general the blush is off the rose in terms of the emotional appeal of computer science as a subject for study, and the drop in interest in IT-flavored computer science courses reflects this.

A lack of interest in technical studies is not what ails US students, just a lack of interesting technical subjects. Enter robotics. Robotics has displaced computer science as the academia's technological glamour queen. Robotics in the US is hot judging by the number of robotics courses being added to grade school, high school, and university curricula. Toss in the robotics technical camps offered dur-

ing school holidays and what you have is a swelling of interest in a decidedly technical subject and one that incorporates not only computer science, but mechanical and electrical engineering, as well.

The 'New' Robotics Market is Not 'Real'

Of course, the US lags Asia in the mobile robotics market. The market is not real, at least not yet – no successes, no business model, and no killer application. At least that's how the story goes. It would seem that the iRobot IPO would have retired this particular truism, but according to robotics naysayers a single IPO does not a revolution make. True, but if you also toss in the massive military robotics market, the commercial Unmanned Aerial/Ground/Underwater Vehicles market, the increasing interest in robotics from toy manufacturers such as Hasbro and Mattel (following the success of Wow Wee Robotics and the UGOBE Pleo announcement), and other initiatives, you can see a groundswell building.

To date, the VCs have been tightfisted with investment dollars for robotics companies and investment, and investment dollars is a sure measure of market reality. It is equally true, however, that you cannot walk across a room at business-oriented robotics conferences without bumping into a venture capitalist.

I do think that the mobile robotics industry is still looking for its 'killer app,' and I fervently believe that more than one will be uncovered. Time and technological advancement make it so. Whether that application is something completely new (the Internet) or a better turn on an existing practice (spreadsheets for ledger books or email for snail mail) has yet to be determined.

The US Does Not Have a Culture of Robotics

This old chestnut is so patently wrong that a rebuttal would hardly seem necessary. In terms of popular culture, we have R2D2, C3PO, Disney World, and the Terminator (played by the now governor of California). We have already touched on the military robotics, the iRobot IPO, new products from US companies entering the market, the addition of robotics to education curricula at all grade levels, and on and on.

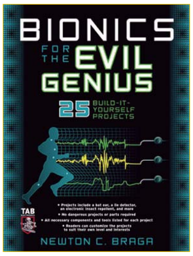
For a country without a culture of robotics, competitions and events seem to be extremely popular (FIRST, BotBall, RoboOlympics, to name but a few). Other robotics events such as the Tetsujin robotics weightlifting contest or robotic gladiatorial combat where 300 pound behemoths go head-to-head, are almost exclusive to the US. We can leave out the argument on whether such remote control robotic warriors are robots at all (they are), or whether they are good for the industry (again, yes), the fact remains that they are rousing good fun and for that reason they draw people by the thousands.

The fact is that the mobile robotics industry is alive and well in the United States. The market is very different from that in Asia and in Europe to be sure, but the overall market is large enough to support any number of national, commercial, and personal approaches for engaging in it. SV

Mind Candy For Today's Roboticist

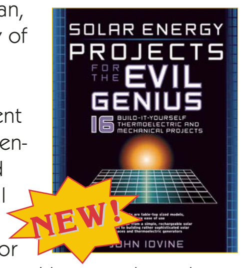
The SERVO Store

Bionics for the Evil Genius
by Newton C. Braga



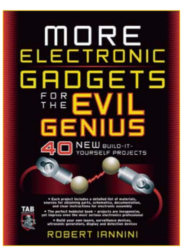
Step into the future (or the past, if you have a touch of Dr. Frankenstein in your soul) with these 25 incredibly cool bionic experiments! Demonstrating how life forms can be enhanced, combined, manipulated, and measured with electronic and mechanical components, these inexpensive projects from internationally renowned electronics guru Newton Braga provide hours of fun and learning. Totally safe, *Bionics for the Evil Genius* guides you step-by-step through 25 complete, intriguing, and low-cost projects developed especially for this book — including an electric fish, a bat ear, a lie detector, an electronic nerve stimulator, a panic generator, and 20 other exciting bioelectric/mechanical projects! **$24.95**

Solar Energy Projects for the Evil Genius
by John Iovine



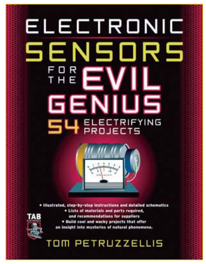
Solar energy is a clean, inexhaustible supply of energy that is being explored more and more given the current state of our dependency on foreign oil and global environmental concerns. This is an easy, fun reference for building 16 practical and inexpensive solar energy projects. All projects are table-top sized models for ease of use and can be scaled larger, if necessary. **$24.95**

MORE Electronic Gadgets for the Evil Genius
by Robert E. Iannini



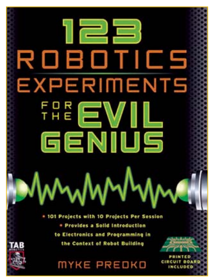
This much anticipated follow-up to the wildly popular cult classic *Electronic Gadgets for the Evil Genius* gives basement experimenters 40 all-new projects to tinker with. Following the tried-and-true Evil Genius Series format, each project includes a detailed list of materials, sources for parts, schematics, documentation, and lots of clear, well-illustrated instructions for easy assembly. Readers will also get a quick briefing on mathematical theory and a simple explanation of operation. **$24.95**

Electronic Sensors for the Evil Genius — 54 Electrifying Projects
by Thomas Petruzzellis



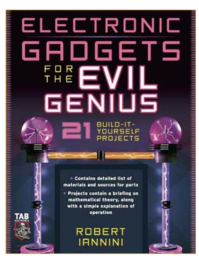
Nature meets the Evil Genius via 54 fun, safe, and inexpensive projects that allow you to explore the fascinating and often mysterious world of natural phenomena using your own home-built sensors. Each project includes a list of materials, sources for parts, schematics, and lots of clear, well-illustrated instructions. Projects include rain detector, air pressure sensor, cloud chamber, lightning detector, electronic gas sniffer, seismograph, radiation detector, and much more! **$24.95**

123 Robotics Experiments for the Evil Genius
by Myke Predko



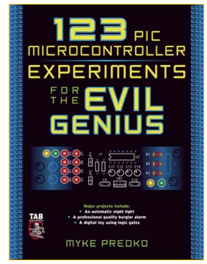
If you enjoy tinkering in your workshop and have a fascination for robotics, you'll have hours of fun working through the 123 experiments found in this innovative project book. More than just an enjoyable way to spend time, these exciting experiments also provide a solid grounding in robotics, electronics, and programming. Each experiment builds on the skills acquired in those before it so you develop a hands-on, nuts-and-bolts understanding of robotics — from the ground up. **$25.00**

Electronic Gadgets for the Evil Genius
by Robert Iannini



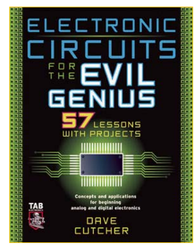
The do-it-yourself hobbyist market — particularly in the area of electronics — is hotter than ever. This book gives the "evil genius" loads of projects to delve into, from an ultrasonic microphone to a body heat detector, all the way to a *Star Wars* Light Saber. This book makes creating these devices fun, inexpensive, and easy. **$24.95**

123 PIC Microcontroller Experiments for the Evil Genius
by Myke Predko



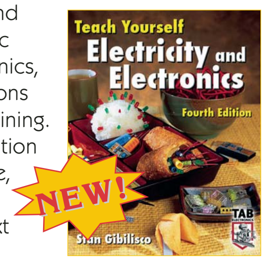
Few books take advantage of all the work done by Microchip. *123 PIC Microcontroller Experiments for the Evil Genius* uses the best parts, and does not become dependent on one tool type or version, to accommodate the widest audience possible. Building on the success of *123 Robotics Experiments for the Evil Genius*, as well as the unbelievable sales history of *Programming and Customizing the PIC Microcontroller*, this book will combine the format of the evil genius title with the following of the microcontroller audience for a sure-fire hit. **$24.95**

Electronic Circuits for the Evil Genius
by Dave Cutcher



Cutcher's 57 lessons build on each other and add up to projects that are fun and practical. The reader gains valuable experience in circuit construction and design and in learning to test, modify, and observe results. Bonus website **www.books.mcgraw-hill com/authors/cutcher** provides animations, answers to worksheet problems, links to other resources, WAV files to be used as frequency generators, and freeware to apply your PC as an oscilloscope. **$24.95**

Teach Yourself Electricity and Electronics — Fourth Edition
by Stan Gibilisco

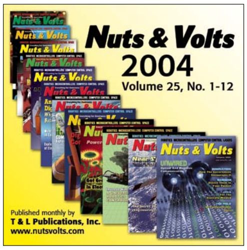


Learn the hows and whys behind basic electricity, electronics, and communications without formal training. The best combination self-teaching guide, home reference, and classroom text on electricity and electronics has been updated to deliver the latest advances. Great for preparing for amateur and commercial licensing exams, this guide has been prized by thousands of students and professionals for its uniquely thorough coverage ranging from DC and AC concepts to semiconductors and integrated circuits. **$34.95**

To order call 1-800-783-4624 or go to our website at www.servomagazine.com

Mind Candy For Today's Roboticist

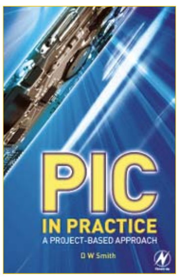
Nuts & Volts CD-Rom



Here's some good news for *Nuts & Volts* readers! Starting with the January 2004 issue of *Nuts & Volts*, all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 25, issues 1-12, for a total of 12 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

PIC in Practice: A Project-based Approach — Second Edition

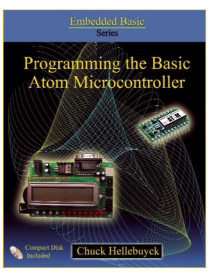
by David W. Smith



PIC in Practice is a graded course based around the practical use of the PIC microcontroller through project work. Principles are introduced gradually, through hands-on experience, enabling students to develop their understanding at their own pace. The book can be used at a variety of levels and the carefully graded projects make it ideal for colleges, schools, and universities. Newcomers to the PIC will find it a painless introduction, while electronics hobbyists will enjoy the practical nature of this first course in microcontrollers. **$29.95**

Programming the Basic Atom Microcontroller

by Chuck Hellebuyck



Through his unique way of making the complicated understandable, Chuck takes the reader through the inner workings of the Basic Atom by explaining the Microchip PIC Microcontroller and its roll in the Atom module. From there, Chuck explains the various PIC based Basic Atom modules and how to use the Basic Atom compiler. Chuck then delivers 13 projects the reader can build and learn from. The reader can then use this knowledge to develop their own Basic Atom projects. **$39.95**

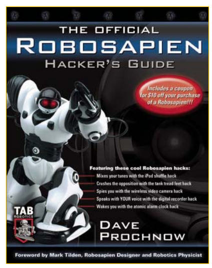
Check out our online bookstore at www.servomagazine.com for a complete listing of all the books that are available.

SERVO CD-Rom

Are you ready for some good news? Starting with the first *SERVO Magazine* issue — November 2003 — all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 1, issues 11-12 and Volume 2, issues 1-12, for a total of 14 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

The Official Robosapien Hacker's Guide

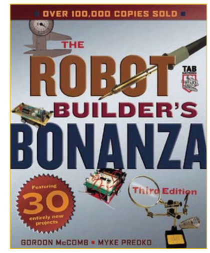
by Dave Prochnow



The Robosapien robot was one of the most popular hobbyist gifts of the 2004 holiday season, selling approximately 1.5 million units at major retail outlets. The brief manual accompanying the robot covered only basic movements and maneuvers — the robot's real power and potential remain undiscovered by most owners — until now! This is the official Robosapien guide — endorsed by WowWee (the manufacturer) and Mark Tilden (the designer). This timely book covers possible design additions, programming possibilities, and "hacks" not found any place else. **$24.95**

Robot Builder's Bonanza Third Edition

by Gordon McComb / Myke Predko



Everybody's favorite amateur robotics book is bolder and better than ever — and now features the field's "grand master" Myke Predko as the new author! Author duo McComb and Predko bring their expertise to this fully-illustrated robotics "bible" to enhance the already incomparable content on how to build — and have a universe of fun — with robots. Projects vary in complexity so everyone from novices to advanced hobbyists will find something of interest. Among the many new editions, this book features 30 completely new projects! **$27.95**

Linux Robotics

by D. Jay Newman

If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need, plus complete training in doing original programming. **$34.95**

$20 each
15% off if you buy two or more.

From HomoSapien to RoboSapien

Before R2D2 there was R1D1

SERVO Magazine T-Shirts

Get your very own limited-edition *SERVO Magazine* T-shirt. Shirts come in sizes S, M, L, and XL, and are available in either black or white. All shirts are 100% pre shrunk cotton.

Group Robot Projects

Or "Why Hasn't Our Robot Managed to Get That Beer Yet?"

by Alex Brown

Are we having fun yet? Well, of course we are! The field of robotics has something for everybody (well, our kind of everybody anyway): hardware, software, electronics, etc. We have clubs and contests and many of us are bringing our own mechanical Frankenstein's monsters to life or are even working on commercial products.

Robotics is a hobby that can last your whole life. Looking at the recent history of robotics, it is apparent that it is a *very* complex field and that progress is made very slowly. For hobbyists, this is great! We actually have the chance to participate in the development of what will surely be a major industry in the near (or far) future. Robotics encompasses many different technologies. It is rare that one person can be expert in all the necessary fields. Fortunately, you don't have to be an expert in all fields to create a robot and there are many books and websites on which you can learn the science and engineering behind robot construction.

So, what is the difference between what we are doing as hobbyists and what academia and industry are doing in robotics? Hobbyist robots tend to be fairly small scale and built for specific tasks (e.g., line followers, firefighters, Robo-Magellan). On the other hand, the research robots of academia and industry tend to be larger scale and designed for more general behaviors (e.g., MIT's Cog and Kismet, Honda's Asimo, Sony's Quio). What is the reason that the pros can create such impressive robots and our own creations seem to have less capability? Well, certainly the pros can throw a lot more money into a project than the average amateur. There isn't much we can do to match them there; but a second reason is that those robots were created by teams of people, while most hobbyist robots are the work of one person.

So, why don't we have more robots on the level of what the professionals produce? A major reason is the magnitude of the task. The most impressive robot projects are done by teams of people, each working in their own areas of expertise. This is because there is a *lot* of hardware and software design and development necessary to create a sophisticated robot. For a single person to do an entire robot means either a lifelong job or keeping the project relatively simple. When robots are built by teams, it is possible to have "experts" working on each area of design. This accomplishes two things: the amount of knowledge that goes into each part of the robot can be higher due to specialization and the robot can be completed faster since work goes on in parallel.

There seem to be two general categories of people who want to work in robotics. There is the generalist who wants to do it all; to learn and design all aspects of a robot in their own way. This is actually the classic hobbyist and certainly the category in which I fit. These people can turn out some pretty impressive robots but are generally limited in how much time they have to put into research, design, and development.

The second category is people who are interested in a particular aspect of robotics (e.g., artificial life/personality, vision, navigation and mapping, manipulation, etc.). These people want to work on their area of interest and need a robot with which to do it, but don't want to spend the time learning everything else about robotics to design their own robot. By specializing, they will tend to know much more in their particular area than the generalist. Of course, without a robot, the specialists may never get to try their ideas in a real mobile robot environment. And, since they may be less expert in many of the other areas necessary, they may never actually get around to building one.

Considering the state-of-the-art, we hobbyists, as a group, should have the ability to create robots that are the equal to or better than anything being built in industry or academia. Among our numbers are people with equal expertise in all the necessary fields. Certainly, we have mechanical, electronics, and software designers. And we have people interested in the unique fields associated with robotics, such as control, navigation, artificial intelligence, vision, speech and voice recognition, etc.

Okay, we're not likely to build an Asimo or Cog because the cost of the components is just too high. But the largest part of robotics is developing sophisticated software to perform the necessary tasks. This is good since it means that once a robot platform is obtained along with the necessary software tools, most of the hobby is relatively low cost (for hobbyists, I'm assuming that the vast number of hours invested is zero cost). We just have to figure out how to keep the hardware costs down and develop the heck out of the software.

What do we really want in a robot? My list includes: mobility, manipulation, sensing, autonomous control, and the ability to add numerous AI functions to play with. I think that we already have mechanical solutions to accomplish all of these. Not optimized solutions, but good enough to perform amazing tasks *if* we just had the right software.

Moving on wheels will take a robot almost anywhere in a house or office environment, except perhaps for those pesky stairs. We seldom see robots with arms and grippers; but I suspect this is more because the software to use them effectively is so difficult rather than because it is technically difficult to build the hardware. We have a variety of sensors, such as IR range, sonar, heat, and vision, which could guide a robot through its environment *if* we just had the complex software to do so. And we have accomplished autonomous operation easily; what is more difficult is to add intelligence, learning, problem solving, and personality.

Take the DARPA Grand Challenge as an example. Almost all the hardware used to achieve this very difficult task was available off the shelf. In the first year, while many teams had put the hardware package together, few had software adequate to the task. In the second year, devoted primarily to software development, most teams performed with much better control and a number of entries actually completed the task.

Asimo is very expensive because Honda decided to take the approach of making the robot walk. Experimenting in robotics does not require a walking robot. A robot on wheels or tracks can be highly mobile within a slightly more limited environment at a reasonable cost. Actually, I doubt they let Asimo go outside much; so his environment is probably also limited.

My point is that developing new and unique hardware capabilities is nice and advances the state-of-the-art, but we already have sufficient hardware designs and most of them can be constructed quite economically. What we need is a *lot* more software development.

A number of people have recognized this problem and are promoting group robot projects; the idea being that individuals can contribute components (hardware or software) in their area of expertise and all the other participants can either use that component, not use it, or create their own "better" version. Of course, in order to provide the visibility for continuous improvement, the designs must be "open source" as far as possible.

There are many websites which provide free open-source software in many areas of robot design, including operating systems, sensors, control, navigation, AI, speech and voice recognition, and much more. Take a look at the selected few listed below:

http://rossum.sourceforge.org

The Rossum project (**rossum.sourceforge.org**) was started by Gary Lucas about six years ago with the goal of collecting, developing, and distributing open source software for robotics. It provides a robot simulator, various papers and software for motion control, localization, etc.

http://cmusphinx.sourceforge.net/html/cmusphinx.php

Carnegie Mellon provides open source speech recognition and synthesis software.

www.alicebot.org

The ALICE Artificial Intelligence foundation provides artificial personality software.

http://carmen.sourceforge.net

Carnegie Mellon provides open source navigation software.

These sites contain a lot of open source code which may be useful in designing your robot, however, it can still be a daunting task to take all these components and figure out how to integrate them and to design a mechanical platform for the robot to run on. Few people want to take on this big a job, basically because it is likely to take a long time to get their robot running.

A few web projects attempt to provide a complete design for an operating robot; with hardware platform design, basic control, operating system, sensors, speech and voice recognition, AI, etc., all in a complete and tested design. With this approach, a newcomer can get a robot operating with a reasonable amount of work and then have a starting point to modify, add to, whatever. This is similar to developing software where it is usually *much* easier to take an operating program and modify it than to develop a set of code from scratch.

Several such projects exist now. Take a look at the ones listed below:

www.symbio.jst.go.jp/PINO/

The Open Pino platform provides hardware and electronics design for an RC servo driven walking robot (or you can buy the platform for just $30,000!). This project started in 2001, but doesn't appear to have much on-line software available yet.

http://cs.gmu.edu/~eclab/projects/robots/flockbots/pmwiki.php?n=Main.Home

This website provides a design for a small robot, "flockbot," suitable for multiagent research, robotics education, and other tasks. Their goal is to get as much functionality as possible from $800 per robot and to document the results to make it easier for others to do the same. The robot is a roughly circular two-platform differential-drive mobile robot seven inches in diameter. It is constructed nearly entirely of off-the-shelf material. The website has a parts list and plans for most of the mechanical design. It also has software source code in C, Java, and Python.

http://oap.sourceforge.net/

The Open Automation Project, started by Dafydd Walters in 2001

provides a parts list and plans to construct a differential-drive base for a robot and provides some software to get started. It uses a PC as the main computer and Linux as an operating system. The goal is a robot which costs under $2,000 to construct. The prototype has a 12 x 12 inch base and is approximately 24 inches high and would provide a more capable platform for research and development than the "flockbot" above.

http://leafproject.org

The Leaf project began in 2003 and provides both hardware plans and complete software to build an operating robot. The average robot being built is 18 to 20 inches in diameter and 30 to 60 inches high. It uses a microcontroller to handle real time tasks like motor control and sensors, and a PC running Windows XP for higher level control including navigation, vision, speech synthesis and recognition, and an artificial personality. It is coded in C, C++, and Lisp. The robot can be constructed for under $2,000 or a bit more, depending on your choice of laptop. Seven Leaf robots are running and a number of others are under construction.

It is common folklore in robot clubs that team projects usually do not succeed. Most such projects start from scratch and assume that a mechanical expert, an electronics expert, and a programmer are going to work together. I haven't worked on such a project, but I would suspect that many bog down on the near-infinite number of decisions necessary to come up with a practical design.

A far easier approach is to start with an already working hardware/software design and enhance it. As long as the starting point has the basic capabilities and provisions to add other functions of interest, the members can have working robots quickly and can then immediately jump into developing their own areas of interest.

On the Leaf Project, we have found that a complete working design results in a robot which can be easily assembled by other team members and quickly made operative. We find that once a team member has a robot running, they immediately want to start making changes to it. We have people working on enhancing the navigation capabilities, adding more functionality to the artificial personality, adding more ability to interact with the outside world (e.g., X10 household controls), improving vision capabilities, adding an arm, and even modifying an "Alive Chimpanzee" as a head.

So, I'm suggesting that once you have worked on relatively simple robots and understand the basics, that you then consider getting involved with a group robot project that matches your interests. Currently, the selection of such groups is somewhat limited; so secondly, I suggest that people who have already created operating robots with lots of potential for future development document their designs on the web and invite others to participate. I hope that such projects will soon have our robots getting us that beer on command. **SV**

AUTHOR BIO

*Alex Brown spent the better part of 30 years as a system designer of autopilots for commercial aircraft "giant flying robots." Since retiring, he has spent vast numbers of hours working on hobby robots and having much more fun! He is one of the founding members of the Leaf project and has his own website at **abrobotics.com** He can be reached at rbirac@cox.net*

Then and NOW

THE UMI R-THETA ROBOT BECOMES THE POPULAR RTX TEACHING ROBOT ARM

by Tom Carroll

This column is about advances in different types and manufacturers of robots made over the years, in particular, robots of interest to experimenters. This month, I'd like to talk about a different configuration of robot, as well as a European company that specializes in this type of robot. Readers of *SERVO* and other magazines have heard me talk about SCARA robots in some of my other articles. That's an industrial robot acronym meaning Selective Compliant Articulated Robot for Assembly, or as some prefer, Selective Compliant Assembly Robot Arm.

The SCARA robot usually has 4-6 axes. The kinematics is much like a human arm with the first joint being referred to as the shoulder and the second as the elbow. Unlike a "typical" robot arm, the SCARA arm moves in a horizontal plane with all the axes vertical. This configuration of robot is the most popular type for assembling circuit boards and for other small assembly operations. It has the advantage of low arm mass, thus allowing high operational speeds.

I had heard about the new SCARA robots being presented at the various RI-SME robot shows that I've attended, but gave them little notice. The new Japanese robot companies that were springing up at that time all had lines of various SCARA robots for assembly tasks. After delivering a paper on an entirely different subject (space robotics) at one of the RI-SME symposiums in 1984, I was approached by Geoff Henny of Universal Machine Intelligence of London. He was trying to develop a mobile robot with a SCARA arm and invited me to go to London and help the company with the design process.

Geoff showed me a few sketches and drawings of his company's designs of a mobile robot with a single SCARA arm. I did a bit of research on this type of robot arm before my first of several trips to the UK. The SCARA configuration was simpler than "Cartesian," "polar," and "revolute" configurations commonly used in industrial applications and was somewhat similar to the "cylindrical" configuration. As with most industrial robots, the castings were beefy and the Z-axis was driven up and down by a lead screw for accuracy and large load capacity. I felt that this type of arm would be a good choice and the least expensive to construct for a mobile robot.

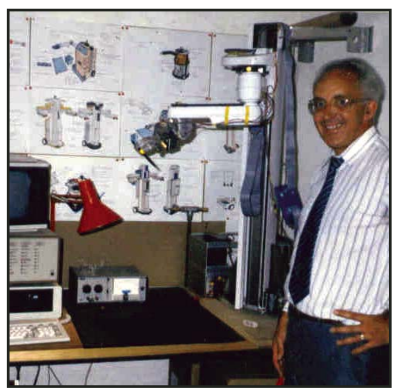
When I got to their offices crammed into an old historic building, I was amazed at the advances that they had accomplished in the science of robotics. Remember, this was the summer of 1984 and the computers back then were the "XT" and "AT" styles of 286 and 386 machines. Tim Jones, Chief Engineer of UMI, proudly showed me a prototype of the R-Theta mobile robot with a SCARA arm that traversed the entire left side of the robot. Figure 1 shows the clean lines of the 38-inch tall robot that sadly never made production status. Tim and the rest of UMI's management team soon realized that control of a mobile robot's base required even more processing power than the control of the arm and that integration of numerous sensors was required for simple navigation.

From the beginning, UMI realized that the separate arm could be marketed for many different purposes, from use in classroom labs to rehabilitation and enabling of disabled persons. This is the path that UMI followed after discarding the idea to further develop the R-Theta mobile platform. Figure 2 shows a prototype stand-alone arm that was in development while I was at UMI in 1984. The arm was later designated as the RTX arm.

Figure 3 is a drawing of the drive mechanisms and the linear slideway track upon which the arm's shoulder traverses. Notice that the arm is driven up and down by a toothed timing belt,

Figure 1. A prototype of the R-Theta mobile robot with a SCARA arm.

Figure 2. A 1984 prototype for a UMI's stand-alone arm.



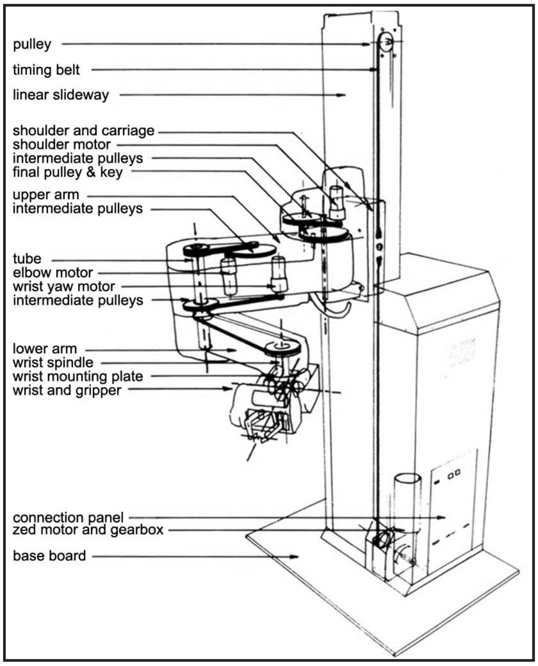


Figure 3. The drive mechanisms and linear track for the arm's shoulder.

Information on the RTX/RT Arms is Available At:
Oxford Intelligent Machines (OxIM)
12 Kings Meadow,
Osney Mead Industrial Estate
Oxford, OX2 0DP, UK
Tel: **+44 (0) 865 204881**
Fax: **+44 (0) 865 204882**
Contact: Dr. Peter Davey

a distinct departure from traditional robot drive systems that use ball-screw drives or similar. This is the same Z-axis drive system that was used in the R-Theta mobile robot design. This design detail concerned me a bit while I was there, but I realized that it is significantly less expensive to manufacture a belt-drive system, despite the lowered accuracy and smaller payload capacity.

Just as Dean Kamen's two-wheeled robot balancing system used in his Independence IBOT Mobility System later became the world-famous Segway Transporter, UMI found better-suited markets for its product in other areas. UMI first looked at light industrial applications for the arm system, but the flimsy arm segments were not adequate for the rigidity requirements of most pick-and-place or assembly tasks. When university labs and robotics courses began to purchase the arm in 1986, UMI knew it had a winner on its hands. Extreme accuracy, speed of motion, and payload capacity are not required for students and researchers to understand robot programming and to develop specialized code for a particular application. Universities in the UK and throughout Europe, as well as in the US, purchased the arm for educational and laboratory uses.

The RTX arm has seven motors (including the gripper), which give it seven degrees of freedom. The RT100 is an up-rated industrial version of the RTX educational robot. The RT100 is designed to be controlled from a personal computer (PC) via an RS232 serial interface. A few of these machines were sold but not enough to satisfy the "bean counters." Towards the end of the 80's, UMI went through some changes as part of a larger umbrella organization. The operation moved over to Oxford, England where the company had a robot welding torch manufacturing facility. A new company, founded in January 1990 as Oxford Intelligent Machines Ltd., was formed and later changed its name to OxIM Ltd.

Rehabilitation robotics became the application that made the RTX, and later, the R100, R100+, R200, and R200+ arms famous. The arms were attached to a table or wall and interconnected to a basic PC through a serial interface and allowed researchers to develop a workstation to assist handicapped and disabled individuals in daily tasks. The workstation was developed for a specific person's needs and contained literature and other materials that this person might need during a typical workday. Verbal or other types of commands to the computer instructed the arm to retrieve or move objects as required.

A survey in 1989 indicated that 38% of all workstation-based rehabilitation robotics projects worldwide were using the RTX arms. In the US, one of the first RTX robot arms was used at Boeing in Seattle to assist one of their disabled programmers. The SCARA geometry of the robot's arm has dimensions approximating those of an adult. Industrial robot speed, accuracy, and capacity are not required to assist a person with their daily needs.

Cambridge University and other European universities worked with OxIM to develop more sophisticated workstations in the late 90's. The RAID or Robot to Assist the Integration of the Disabled workstation was developed, and several versions have been implemented around the world. The RAID 2 Workstation featured the latest RT200 robot arm and was programmed using CURL, or Cambridge University Robot Language.

Several university researchers have looked at a series of low-cost sensors such as proximity, distance, force, and slip sensors for use in end-effectors (grippers) for the RTX robot. With greater computer "intelligence" available, autonomy and vision sensors have been implemented.

The arm can access books, music CDs, CD-ROMs, and documents from a specially constructed shelf system using specialized end effectors. Book pages can be turned forward and backward. The uses of the RTX/RT robot arms for rehabilitation purposes are virtually unlimited and companies in the UK are determined to stay in the forefront of robotics. SV

Advertiser Index

All Electronics Corp. 29, 88
Critical Velocity 88
CrustCrawler 25
Dimension Engineering 39
Futurlec 88
Garage Technologies 88
Hitec 41
Hobby Engineering 9
Jameco Robot Store 2
Lorax Works 29, 88
Lynxmotion, Inc. 62
Net Media 91
Parallax, Inc. ....Back Cover
Pololu Robotics & Electronics 83
ROBO Business 18
ROBOlympics/Tetsujin 2006 3
RoboteQ 7
Robot Power 71
Robot Shop 43, 88
Solarbotics..17
Solutions Cubed 75
Technological Arts 88
Vantec 22
Yost Engineering 13